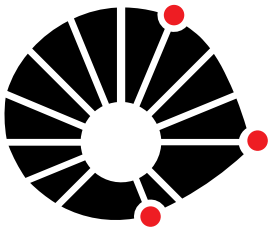


UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

DARWIN GREGORIO VILLAR SALINAS

On linear block codes: classification and estimation of bounds for weight hierarchy of codes

Sobre códigos lineares de blocos: classificação e estimativa de cotas para hierarquia de pesos de códigos.

Campinas

2022

Darwin Gregorio Villar Salinas

On linear block codes: classification and estimation of bounds for weight hierarchy of codes

Sobre códigos lineares de blocos: classificação e estimativa de cotas para hierarquia de pesos de códigos.

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Matemática.

Thesis presented to the Institute of Mathematics, Statistics and Scientific Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Mathematics.

Supervisor: Marcelo Firer

Este trabalho corresponde à versão final da Tese defendida pelo aluno Darwin Gregorio Villar Salinas e orientada pelo Prof. Dr. Marcelo Firer.

Campinas

2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

V712o

Villar Salinas, Darwin Gregorio, 1985-
On linear block codes : classification and estimation of bounds for weight hierarchy of codes / Darwin Gregorio Villar Salinas. – Campinas, SP : [s.n.], 2022.

Orientador: Marcelo Firer.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Códigos corretores de erros (Teoria da informação). 2. Peso generalizado de Hamming. 3. Dualidade (Matemática). 4. Análise espectral - Programas de computador. 5. Representações monomiais. I. Firer, Marcelo, 1961-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Sobre códigos lineares de blocos : classificação e estimativa de cotas para hierarquia de pesos de códigos
Palavras-chave em inglês:
Error-correcting codes (Information theory)
Generalized Hamming weight
Duality theory (Mathematics)
Spectrum analysis - Computer programs
Monomial representations
Área de concentração: Matemática
Titulação: Doutor em Matemática
Banca examinadora:
Marcelo Firer [Orientador]
Javier Alfonso de la Cruz Cantillo
Luciano Panek
Sueli Irene Rodrigues Costa
Roberto Assis Machado
Data de defesa: 31-03-2022
Programa de Pós-Graduação: Matemática

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-2617-3191
- Currículo Lattes do autor: http://lattes.cnpq.br/6021747470585457

Tese de Doutorado defendida em 31 de março de 2022 e aprovada

pela banca examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). MARCELO FIRER

Prof(a). Dr(a). JAVIER ALFONSO DE LA CRUZ CANTILLO

Prof(a). Dr(a). LUCIANO PANEK

Prof(a). Dr(a). SUELI IRENE RODRIGUES COSTA

Prof(a). Dr(a). ROBERTO ASSIS MACHADO

A Ata da Defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-Graduação do Instituto de Matemática, Estatística e Computação Científica.

Para las hermosas mujeres que Dios ha colocado en mi vida para ayudarme a ser un mejor hombre, Arellys, Abby, Hannah y Sonia. Las amo, pero nunca tanto como Cristo nos amó y nos ama.

Acknowledgements

I wish to thank first God, for his patience, mercy, love and forgiveness He has had with me every step of my life. Because certainly if it was not for Him I would not be alive and much less would I have the chance to get to this point. To my beautiful wife, who I wish her name could be written in my diploma. She has been a blessing in my life and an instrument given by God to help me being shaped in the likeness of Jesus. To my dear and beloved daughters, who have suffered along side with us, sacrificing part of their time with me so that I could take this project ahead. To my mom, dad and siblings, my aunt Rosalía and uncle Dale, who always supported me from the beginning and even today are always there whenever needed. To the body of Christ represented by the IBBG(Igreja Batista em Barão Geraldo) for being our family while being away from home, the Biblical Church in Aachen for their love and support even in the darkest and coldest hour and more recently IBDC(Iglesia Bíblica de la Ciudad) for their love in Christ even without previously knowing us.

I also wish to express my gratitude to my supervisor, Prof. Dr. Marcelo Firer, who has also given me unconditional support from the beginning. His expertise and knowledge have been crucial to take this thesis ahead. To all the professors along the way who have helped me understand better the world of Mathematics and how to use it to make our world a better place, Prof. Dr. Ismael Gutiérrez, Prof. Dr. Javier de la Cruz, Prof. Dr. Stefka Bouyuklieva, Prof. Dr. Gabriele Nebe, Prof. Dr. Wolfgang Willems, Prof. Dr. Mahendra Panthee, Prof. Dr. Luciano Panek and so many others. Including the members of my evaluation committee not mentioned before, Prof. Dr. Sueli Costa and Prof. Dr. Roberto Assis Machado.

My gratitude to Brasil and the whole Unicamp community for receiving me just as if I were one of your own. I am especially thankful with my friends Attilio and Juliana Ropole, Alicia Danes, the grandparents Brasil gave to my daughters; Mónica, Sebastián, Carlos Fabián, Adriana, Deimer, Lina, Yessica, Rogelio, Ricardo, Catherine, Pascual, Janeth, Nayara and family, each one of you were angels God put in the right place at the right time. Thank you.

This work has been supported partially by Colciencias, Univesp, SAE-Unicamp and DAAD. Without your financial support this work would not have been possible.

Mas la senda de los justos es como la luz de la aurora,
Que va en aumento hasta que el día es perfecto. Proverbios 4:18

Resumo

Neste trabalho introduzimos figuras de mérito que permitem a comparação da capacidade de correção de erros de códigos com parámetros clássicos semelhantes. Calculam-se alguns limitantes para elas usando o princípio de Inclusão-Exclusão. Obtiveram-se expressões fechadas no caso de códigos perfeitos. Tudo considerando os pesos generalizados de Hamming. Conjecturamos que o espectro do código é um conjunto completo de invariantes de códigos, sendo assim uma potencial ferramenta de classificação. Finalmente, também foram classificados alguns códigos extremais Tipo III, para os quais dois novos códigos extremais foram encontrados sendo feita uma generalização usando representações monomiais.

Palavras-chave:Códigos corretores de erros, Peso generalizado de Hamming, Dualidade, Análise espectral-Programas de computador, Representações monomiais.

Abstract

In this work we introduce a couple of figures of merit that allow the comparison of the error correction capacity of codes with similar classical parameters. Some bounds for them are calculated by means of the Inclussion-Exclussion principle. Closed expressions are obtained for perfect codes. All this taking into consideration the generalized Hamming weights. We conjecture the usage of the spectrum of a code as a complete set of invariants of codes, being this way a potential classification tool. Finally, we also classified some extremal Type III codes, for which a couple of new extremal codes were found and a generalization is presented using monomial representations

Keywords: Error-correcting codes, Generalized Hamming weight, Duality, Spectrum analysis-Computer programs, Monomial representations.

Contents

Introduction

Due to the purpose of codes in the transmission of information, it is important to know how many errors a code can correct or detect in a given scenario. In order to be able to decode a message efficiently, one generally works on linear codes. The idea of error correction has been usually associated only to the idea of always being able to correct or detect such number of errors, which is directly related to the concept of minimum distance of the code. But it is also possible to think that the chances a code has to correct a number of errors, beyond the point where the correction process may be ambiguous, depends also on the distribution of weights of all the codewords it has, and not only in the smallest one. This is what motivated us to look for the impact that the generalized weight of the code has in the error correction capabilities it has, giving a practical usage to the concept introduced by (WEI, 1991) further from being just a theoretical definition.

We begin this thesis with the introduction of some basic definitions and results in Chapter 1 that set the basis to work on two problems in coding theory. First the error correction capacity that binary codes have and second working on the classification of linear codes, both binary and ternary, given some particular conditions. It has been noticed that the weight distribution of the code determines its error correction probability, when transmitted over the erasure channel as studied in (SHEN; FU, 2019; DIDIER, 2006; FASHANDI; GHARAN; KHANDANI, 2008; ROSENTHAL; YORK, 1997). Nonetheless, to study their impact when the position of the errors is unknown, makes the task more complex. In this thesis, we found some general bounds for the error correction capacity a code has and closed expressions applied to codes belonging to the family of perfect codes, as presented in Chapter 2. Secondly, we also got to a point in which we noticed the relationship the spectrum of the code has with determining uniquely a code. And it has interesting implications with the second usual problem in coding theory already mentioned, the classification of codes. We have proved, by computational exhaustion, that every code of length $n \leqslant 11$ is uniquely determined up to equivalence by its spectrum. The general statement remains a conjecture, but it is strongly supported by additional conditions as it will be seen in Chapter 3. This is also approached in Chapter 4 but from the perspective of the automorphism group of extremal codes, with fixed parameters.

It was pursued the classification of extremal Type III codes of length 36, 52 and 60 with an automorphism of prime order $p \geqslant 5$, and were obtained during the process, nonequivalent codes to the previously known ones, of length 52 and 60. Finally, in Chapter 5 a generalization of the construction for the case of length 60 is presented, making use of monomial representations. This is already published in (NEBE; VILLAR, 2013) and a recent paper in the journal Mathematics (BOUYUKLIEVA; CRUZ; VILLAR, 2022).

1 Preliminaries

In this chapter some basic definitions are introduced, in order to understand the results presented subsequently. Concepts related to the Hamming distance, duality, support of codes, action of groups and generalized Hamming weights. (WEI, 1991; JOSHI, 1989; HUFFMAN; PLESS, 2010)

1.1 Generalities on linear codes

Let $\mathbb{F}_q$ be the Galois field or finite field with q elements and $n \in \mathbb{N}$. A k-dimensional vector subspace $C \leqslant \mathbb{F}_q^n$ is called an $[n,k]_q$-**linear code** over $\mathbb{F}_q$. We call the elements of C **codewords** and if $q = 2$ or $q = 3$, we then say that C is a **binary** or **ternary code**, respectively, in general q**-ary.** The parameter n is the **length** of C.

For $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$ we define the **Hamming distance** between $\mathbf{x}$ and $\mathbf{y}$

$$\mathrm{d}(\mathbf{x}, \mathbf{y}) := |\{j : 1 \leqslant j \leqslant n,\ x_j \neq y_j\}|.$$

Theorem 1.1.1. *The Hamming distance* d *is a metric. Then for every* $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_q^n$ *the following holds:*

1. $\mathrm{d}(\mathbf{x}, \mathbf{y}) \geqslant 0$ *and* $\mathrm{d}(\mathbf{x}, \mathbf{y}) = 0$ *if and only if* $\mathbf{x} = \mathbf{y}$,
2. $\mathrm{d}(\mathbf{x}, \mathbf{y}) = \mathrm{d}(\mathbf{y}, \mathbf{x})$ *(Symmetry),*
3. $\mathrm{d}(\mathbf{x}, \mathbf{y}) \leqslant \mathrm{d}(\mathbf{x}, \mathbf{z}) + \mathrm{d}(\mathbf{z}, \mathbf{y})$ *(Triangular inequality).*

Besides, d *is also invariant under translations. This is, for every* $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_q^n$ *holds that*

$$\mathrm{d}(\mathbf{x} + \mathbf{z}, \mathbf{y} + \mathbf{z}) = \mathrm{d}(\mathbf{x}, \mathbf{y}).$$

Proof. This result is well-known, and the details can be seen in (JOSHI, 1989, Th. 1.5 Ch. 3), for example. We only sketch the proof here. The non-negativity and symmetry are immediate. Let $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_n)$, $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{F}_q^n$. For the

triangular inequality, note that if $x_j \neq y_j$, then $x_j \neq z_j$ or $y_j \neq z_j$. And so we have the statement proved.

On the other hand,

$$\begin{aligned} \mathrm{d}(\mathbf{x},\mathbf{y}) &= |\{j : x_j \neq y_j,\ j = 1,\ldots,n\}| \\ &= |\{j : x_j + z_j \neq y_j + z_j,\ j = 1,\ldots,n\}| \\ &= \mathrm{d}(\mathbf{x}+\mathbf{z},\mathbf{y}+\mathbf{z}). \end{aligned}$$

□

Another classical parameter of a code C is its **minimum distance**, denoted by $\mathrm{d}(C)$ and it is defined by the following rule: if $|C| > 1$, then

$$\mathrm{d}(C) := \min\{\mathrm{d}(\mathbf{x},\mathbf{y}) : \mathbf{x},\mathbf{y} \in C,\ \mathbf{x} \neq \mathbf{y}\}.$$

and if $|C| = 1$, then $\mathrm{d}(C) := 0$. It is based on this parameter that many of the decoding algorithms are constructed, making it very important.

If C is an $[n,k]$-linear code over $\mathbb{F}_q$ and has minimum distance $\mathrm{d}(C) = d$, we say that C is an $[n,k,d]$-code over $\mathbb{F}_q$ or simply we say C is an $[n,k,d]_q$-code. The triplet $[n,k,d]$ is called the **main classical parameters** of the code C.

We define the **Hamming weight** of $\mathbf{x} \in \mathbb{F}_q^n$, represented by $\mathrm{wt}(\mathbf{x})$ as the number of components different from zero in $\mathbf{x}$. As well, the **minimum weight** of $C \leqslant \mathbb{F}_q^n$, denoted by $\mathrm{wt}(C)$ is defined as follows: If $C \neq \{0\}$, then

$$\mathrm{wt}(C) := \min\{\mathrm{wt}(x) : 0 \neq x \in C\}.$$

If $C = \{0\}$, then $\mathrm{wt}(C) := 0$. As a consequence ofTheorem 1.1.1 which states the invariance under translation of d, we have without difficulty that $\mathrm{wt}(C) = \mathrm{d}(C)$. In this thesis only linear codes will be considered. With the help of the weight function, the minimum distance of a linear code can be determined faster. This, taking into account that when calculating $\mathrm{d}(C)$ it is required to do $\binom{q^k}{2}$ operations, while to calculate $\mathrm{wt}(C)$ we do only $q^k - 1$. As k grows the first is much larger than the last one, then the weight simplifies the calculations notably.

The Hamming weight and distance are related by the equalities

$$\mathrm{d}(x,y) = \mathrm{wt}(x-y) \quad \text{and} \quad \mathrm{wt}(x) = \mathrm{d}(x,0).$$

Also, if for a code C of length n, we define $A_i := |\{\mathbf{c} \in C : \mathrm{wt}(\mathbf{c}) = i\}|$, the number of codewords of weight i, then

$$(x) := \sum_{i=0}^{n} A_i x^i$$

is called the **polynomial weight enumerator of** C. By $\mathrm{Sym}(n)$ we denote the symmetric group of order n. Given $\sigma \in \mathrm{Sym}(n)$ and $\mathbf{x} \in \mathbb{F}_q^n$ the action of $\mathrm{Sym}(n)$ on $\mathbb{F}_q^n$ is defined by

$$\sigma(\mathbf{x}) := (x_{\sigma(1)}, \ldots, x_{\sigma(n)}).$$

Equivalently, if we define the associated permutational matrix of σ as

$$B_\sigma = (b_{ij})_{n\times n}, \text{ with } b_{ij} = 1, \text{ if } \sigma(i) = j, \text{ and } 0, \text{ otherwise,}$$

then one can see the action of σ on $\mathbf{x}$ as the matrix product $B_\sigma \mathbf{x}^T$. The **group of linear isometries of** $\mathbb{F}_q^n$, endowed with the Hamming metric, is denoted by $\mathrm{Isom}(n)$. Any element $T \in \mathrm{Isom}(n)$ can be described as the composition $T = A \circ \sigma$, where $\sigma \in \mathrm{Sym}(n)$ and A is a $n \times n$ regular (invertible) diagonal matrix, with coefficients in $\mathbb{F}_q$ or simply put $T = AB_\sigma, A \in GL(n, \mathbb{F}_q)$, the general linear group of all the $n \times n$ invertible matrices with coefficients over $\mathbb{F}_q$. This will be further discussed in Chapter 5.

For C a binary code, if $\sigma(\mathbf{x}) \in C$, for every $\mathbf{x} \in C$, then σ is called an **automorphism** of C. Two codes C, C' are said to be **equivalent**, if there is $T \in \mathrm{Isom}(n)$ such that $T(C) = C$. The set of all the automorphisms of C is the **group of automorphisms** of C and will be written $\mathrm{Aut}(C)$. Using the notation $\mathrm{Mat}(k \times n, \mathbb{F}_q$ for the set of all the $k \times n$ matrices with coefficients over $\mathbb{F}_q$, we define:

Definition. 1.1.1. *Let C be an $[n,k]$-code over $\mathbb{F}_q$.*

1. *If $k \geqslant 1$, then $G \in \mathrm{Mat}(k \times n, \mathbb{F}_q)$ is called a **generator matrix** of C, if*

$$C = \mathbb{F}_q^k G = \{(x_1, \ldots, x_k)G : x_j \in \mathbb{F}_q\}.$$

   In particular, we get $\dim_{\mathbb{F}_q}(C) = \mathrm{Rank}(G)$.

2. *If $k < n$, then $H \in \mathrm{Mat}(n-k \times n, \mathbb{F}_q)$ is called a **parity-check matrix** of C, if*

$$C = \{\mathbf{x} \in \mathbb{F}_q^n : H\mathbf{x}^t = 0\}.$$

On can easily check that

$$\text{Rank}(H) = n - \dim(\ker(H)) = n - \dim(C) = n - k.$$

We say G, a generator matrix of a code C, is in its **standard form** if it can be written as

$$G = (I_k \mid B),$$

here I_k denotes the identity matrix of size $k \times k$. A matrix in its standard form is also in its reduced row echelon form. We remark that every code is equivalent to a code that admits a generator matrix in a standard form. Here, equivalence is considered up to an isometry as indicated before.

Next, we define the dual of a code C. As for euclidean vector spaces, this concept is described using a formal inner product, determined by a bilinear (degenerate) form. For this purpose we define

$$(\mathbf{x}|\mathbf{y}) := \sum_{j=1}^{n} x_j y_j,$$

for $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ vectors in $\mathbb{F}_q^n$.

We define the dual of C, denoted by $C^\perp$, as follows:

$$C^\perp := \{\mathbf{x} \in \mathbb{F}_q^n : (\mathbf{x}|\mathbf{c}) = 0, \ \forall \mathbf{c} \in C\}.$$

If $C \subseteq C^\perp$, then it is said that C is **self-orthogonal**, and if $C = C^\perp$, it is called **self-dual**. It is well known that

$$\dim_{\mathbb{F}_q}(C) + \dim_{\mathbb{F}_q}(C^\perp) = n.$$

Due to this, if C is an $[n, k]$-code over $\mathbb{F}_q$, then $C^\perp$ is an $[n, n-k]$-code. In particular, if C is self-dual, then $n = 2k$.

1.2 Generalized weights

As mentioned in Section 1.1, the main invariants of a code are the length n, the dimension k and the minimal distance d. Here we introduce a generalization of the minimal distance due to Wei (WEI, 1991) which, as we shall see in Chapter 2, is an actual refinement of the minimal distance, as an indicator of the capabilities of a code to correct errors.

The **support** $\mathrm{Supp}(\mathbf{x})$ of a vector $\mathbf{x} \in \mathbb{F}_q^n$ is the set of its non-zero coordinates,

$$\mathrm{Supp}(\mathbf{x}) := \{i | x_i \neq 0\}.$$

The support $\mathrm{Supp}(X)$ of a subset $X \subseteq \mathbb{F}_q^n$ is the set of coordinates that are non-zero for some element of X, i.e.,

$$\mathrm{Supp}(X) := \bigcup_{\mathbf{x} \in X} \mathrm{Supp}(\mathbf{x}).$$

Theorem 1.2.1. *Given a k-dimensional linear code C with generator matrix*

$$G := \begin{pmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_k \end{pmatrix},$$

where $\mathbf{c}_i$ is the i-th row of G. Then,

$$\mathrm{Supp}(C) = \bigcup_{i=1}^{k} \mathrm{Supp}(\mathbf{c}_i)$$

Proof. It is clear that the union of the supports of the codewords that generate the code C is contained in the support of the code itself by definition. Then for the converse we notice that every codeword in C can be written as a linear combination of $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ and, therefore, the support of this codeword is limited to the supports of such generators having then the result. □

This theorem makes the calculation of the support of a linear code much easier, being it limited to the union of the support of generator codewords, k calculations compared to 2^k by the definition in the binary case. Theorem 1.2.1 is used in the algorithm to calculate the support of vector spaces during the classification process further in this thesis.

Now, we could redefine the weight of a vector in terms of its support, since for $\mathbf{x} \in \mathbb{F}_q^n$, $\mathrm{wt}(\mathbf{x}) = |\mathrm{Supp}(\mathbf{x})|$. This gives us a hint of a generalization for the concept of minimum distance of a code C. Then we define the **i-th generalized Hamming distance** $d_i(C)$, as introduced by Wei (WEI, 1991):

$$\mathrm{d}_i(C) := \min\{|\mathrm{Supp}(X)| \ : X \leqslant C, \dim_{\mathbb{F}_q} X = i\}.^{1}$$

[1] $X \leqslant C$ represents X is a subspace of C

Note that, since the support of a nonzero vector equals the support of the 1-dimensional subspace, for $i = 1$ we have that $\mathrm{d}(C) = \mathrm{d}_1(C)$. And as it is also natural for $i = \dim_{\mathbb{F}_q} C = k$, then $\mathrm{d}_k(C) = |\mathrm{Supp}(C)|$ and $\mathrm{d}_k(C) = n$ if and only if the code is non-degenerate, i.e., the only vectors that annihilate the inner product with codewords in C is not the null-vector. From the definition of $\mathrm{d}_i(C)$ we can easily get that $\mathrm{d}_i(C) \leqslant \mathrm{d}_{i+1}(C)$, $i = 1, \ldots, k-1$. So we can summarize the next inequality

$$\mathrm{d}(C) = \mathrm{d}_1(C) \leqslant \ldots \leqslant \mathrm{d}_k(C) = |\mathrm{Supp}(C)|.$$

The sequence of numbers $[\mathrm{d}_1(C), \mathrm{d}_2(C), \ldots, \mathrm{d}_k(C)]$ corresponds to the so-called **Weight Hierarchy of** C.

In fact, it was proved in (WEI, 1991, Theorem 1) that the inequality is strict. Let us consider any $i \in \{2, \ldots, k\}$ and we will check that $\mathrm{d}_{i-1}(C) < \mathrm{d}_i(C)$. Take $M \leqslant C$, with $|\mathrm{Supp}(M)| = \mathrm{d}_i(C)$ and $\dim(M) = i$. For any $j \in \mathrm{Supp}(M)$ take

$$M_j := \{\mathbf{v} \in M : v_j = 0\},$$

i.e., the set of codewords in M with zeros in the j-th coordinate. It is clear $M_j \neq \varnothing$, in fact since $j \in M_j$ is the kernel of the linear operator that maps $v \in M$ to $\mathrm{Proj}_j(v)$, the projection of v on its j-th coordinate. This means M_j is a $(i-1)$-dimensional subspace of M. Then

$$\dim(M_j) = i - 1 \to \mathrm{d}_{i-1}(C) \leqslant |\mathrm{Supp}(M_j)| \leqslant |\mathrm{Supp}(M)| - 1 < \mathrm{d}_i(C)$$

or

$$\mathrm{d}_{i-1}(C) < \mathrm{d}_i(C), \text{ for } i = 2, \ldots, k.$$

2 Error correction capabilities of a code

When transmitting data over a memoryless channel, it is often considered that the transmission is through a symmetric channel, in which the probability a symbol is received correctly or mistakenly is independent of the symbol that was sent. When considering a binary alphabet, the model that represents the channel is the binary symmetric channel represented next. Shannon described the probability of error of a binary symmet-

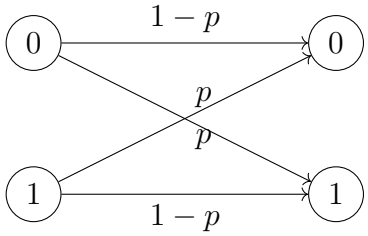


Figure 1 – Binary symmetric channel.

ric channel, where the symbols 0 and 1 had the same chance p of being received wrongly (SHANNON, 1984).

It was proved that for this type of channel, as can be seen in (ROMAN, 1992, Chapter 3), the criteria of **minimum distance**, in which the sent codeword is assumed to be the closest one to the received vector, was equivalent to that of **maximum likelihood decoding**, i.e., considering the transmitted codeword to be the one with the highest conditional probability of getting the received vector provided a codeword was sent. In general, this assumption is not true, as for channels with memory, as indicated in (AZAR; ALAJAJI, 2013). Then it became important to analyze which properties of a code would benefit this kind of decoding with respect to that metric. As seen in (FIRER, 2021), for each channel that has a metric associated, one could deal with this problem considering different metrics.

In our case, we worked based on the Hamming distance to establish the capacity of error correction of a given binary code. One can notice this decoding is completely trustworthy if the number of errors and erasures combined is smaller or equal to the 1**-st packing radius** or the biggest radius for which the balls centered at codewords of C are

disjoint, denoted $R_{pack}(C)$ (we distinguish an order here, considering the weight hierarchy introduced in Chapter 1)

$$\delta_1 := \left\lfloor \frac{\mathrm{d}_1(C)-1}{2} \right\rfloor. \tag{2.1}$$

In other words, suppose that the codeword x is transmitted and the received message is $\mathbf{y} = \mathbf{x} + \mathbf{e}$. Here, $\mathbf{e}$ is the error vector. If $\mathrm{wt}(\mathbf{e}) \leqslant \delta_1$ then, there is a unique codeword closest to y and this is the original message x.

However, it is not possible to ensure that $\mathrm{wt}(\mathbf{e}) \leqslant \delta_1$, if the channel is memoryless. For $\mathrm{wt}(\mathbf{e}) > \delta_1$ a received message may or may not be properly decoded (using either minimal distance or maximum likelihood decoding). One interesting question that comes from this fact, when considering errors beyond the 1-st packing radius, is how could we compare the capacity of a given pair of codes to correct a determined number of errors? For instance, if we have two codes with the same minimum distance, how can we say which one behaves better facing a number of errors bigger than δ_1? As a simple example, let us consider C to be a binary 2-dimensional code generated by the words $\mathbf{e_1} = (1, 0, \cdots, 0)$ and $\mathbf{e_2} = (0, 1, 0, \cdots, 0)$. Let C' be another code, generated by $\mathbf{e_1}$ and $\mathbf{1} = (1, 1, \cdots, 1)$. It is clear that $\mathrm{d}(C) = \mathrm{d}(C') = 1$. We note that both the codes can correct a single error if this error occurs in a coordinate $2 < j \leqslant n$. However, C is not capable of correcting a single error neither on the coordinate $j = 1$ nor $j = 2$, while C' is capable of correcting every single error if it occurs in the coordinate $j = 2$, that is, if $\mathbf{e_2}$ is the error vector. This naive example calls the attention to a possible role of the generalized weights, since $2 = \mathrm{d}_2(C) < \mathrm{d}_2(C') = n$.

This question brings into the actual picture the generalized Hamming weights. The role of these weights to better determine the error correction capabilities of a code is a hard question that has been researched in a few works, of which we quote the following. In 2006, Didier determined bounds for the block error probability over the erasure channel, which he showed is related to the cryptographic algebraic immunity (DIDIER, 2006). Later in 2008, Fashandi, et al., tried to establish the error probability over an erasure channel either with memory or memoryless. They found that it was achieved for MDS codes (FASHANDI; GHARAN; KHANDANI, 2008). Then in 2014 Lemes and Firer took a step further, considering the support's size distribution of subspaces of a given code or spectrum. Analyzing the bounds for the error probability over the erasure channel they

proved some facts about AMDS and MDS codes (LEMES; FIRER, 2014). Lately in 2019, Shen and Fu found a closed expression for this probability also over the erasure channel using the notion of incorrigible sets. They proved how the successful decoding probability is associated to the code's support weight distribution under list decoding and maximum likelihood decoding.

The problem of determining how good a code is in terms of error correction is a difficult one. This, even when considered a transmission over the erasure channel, where the position of the errors is known. That allowed for example the use of list decoding and of incorrigible sets. In our case, under the memoryless binary symmetric channel it is even harder, because the errors can occur imperceptibly anywhere in the codeword. Nonetheless, we work in this thesis on finding a way to check the intrinsic error correction capacity of a code, even when their occurrence is unknown. We take a step further, introducing some new concepts in the next section. These will be the figures of merit that measure the number of vectors in the ambient space that the code can faithfully reconstruct to be a codeword. We'll talk a bit more about this before introducing the definitions themselves.

Given two codes with equal classical parameters. If we consider only their minimum distance, then there is not enough information about them. This is specially true, if we try to compare how good the codes are by themselves, without the influence of the channel. In an approach that is more systematic than the naive example, to determine the capacity of error correction a code has, arises the need to consider the generalized Hamming weights and even the complete, so-called Spectrum of the code. This way, the object to be optimized is simplified by not considering the error probability of transmitting the symbols through the channel. Instead, given a number of errors occur, we consider the proportion of vectors that the code itself can correct.

2.1 $\mathsf{PCD}_{\leqslant e}(C)$ and $\mathsf{PED}_{=e}(C)$

Here, we consider a message $\mathbf{x}$ sent through a channel using a code with minimum distance $\mathrm{d}_1(C)$. We suppose that the received message is $\mathbf{y} = \mathbf{x} + \mathbf{e}$. Taking into consideration Equation 2.1 and the covering radius of C, the smallest radius that allows the complete ambient space $\mathbb{F}_q^n$ to be covered with balls centered in codewords of C,

denoted $R_{cov}(C)$, there are three possible situations that can happen:

1. $\mathrm{wt}(\mathbf{e}) \leqslant \delta_1$;
2. $\delta_1 < \mathrm{wt}(\mathbf{e}) < R_{cov}(C)$;
3. $R_{cov}(C) \leqslant \mathrm{wt}(\mathbf{e})$.

The first case, which includes the case of non-error ($\mathrm{wt}(\mathbf{e}) = 0$) is the best one: the original codeword $\mathbf{x}$ is the closest one to the received message $\mathbf{y}$,

$$\mathrm{d}(\mathbf{y}, C) := \min_{\mathbf{c} \in C} \mathrm{d}(\mathbf{y}, \mathbf{c}) = \mathrm{d}(\mathbf{y}, \mathbf{x}),$$

so the message is always properly corrected.

The last case is the worst one, the original codeword $\mathbf{x}$ is never the closest one to the received message $\mathbf{y}$ ($\mathrm{d}(\mathbf{y}, C) < \mathrm{d}(\mathbf{y}, \mathbf{x})$) so the message is always improperly corrected.

The middle case, when $\delta_1 < \mathrm{wt}(\mathbf{e}) < R_{cov}(C)$ is the case of interest, since some messages may be properly corrected ($\mathrm{d}(\mathbf{y}, C) = \mathrm{d}(\mathbf{y}, \mathbf{x})$) while others not ($\mathrm{d}(\mathbf{y}, C) < \mathrm{d}(\mathbf{y}, \mathbf{x})$) and, an in between case, with some ambiguity, with more than one codeword at distance $\mathrm{wt}(\mathbf{e})$ from $\mathbf{y}$ and one of those is the original $\mathbf{x}$ (there is $\mathbf{x} \neq \mathbf{c} \in C$ such that $\mathrm{d}(\mathbf{y}, C) = \mathrm{d}(\mathbf{y}, \mathbf{x}) = \mathrm{d}(\mathbf{y}, \mathbf{c})$).

Then, for not being defined what happens in general, it is interesting to see what happens in between. What is the situation when the number of errors e is between δ_1 and $R_{cov}(C)$? This is part of the problem we wanted to resolve. We wanted also to obtain tools that would allow codes comparison. In particular, whenever they have similar parameters but different correction capacity. To face this problem, we shall introduce two figures of merit, of a code. The proportion of correct decoding, denoted $\mathrm{PCD}_{\leqslant e}(\mathrm{C})$, and the proportion of exact decoding, denoted $\mathrm{PED}_{=e}(\mathrm{C})$.

In $\mathrm{PCD}_{\leqslant e}(\mathrm{C})$, for each codeword $\mathbf{c}$, we count the number of vectors $\mathbf{y} \in \mathbb{F}_q^n$ at distance $\mathbf{e}$ from $\mathbf{c}$ that are not at the same distance or closer to any other codeword $\mathbf{c}'$ ($\mathrm{d}(\mathbf{y}, \mathbf{c}) \leqslant e$ and $\mathrm{d}(\mathbf{y}, \mathbf{c}') > e$, for all $\mathbf{c}' \in C$) and then it is normalized with respect to the total amount of vectors around $\mathbf{c}$ at a distance e or smaller. It gives a notion of the capacity C has to correct without ambiguity up to e errors.

In contrast to $\mathrm{PCD}_{\leqslant e}(\mathrm{C})$, the other concept, $\mathrm{PED}_{=e}(\mathrm{C})$ counts the proportion of exact decoding or decoding without ambiguity when the number of errors during the transmission is e, then here it is not considered the case less than e errors occur ($\mathrm{d}(\mathbf{y},\mathbf{x}) = e$). We will see that in this particular scenario we have something to say when compared to the packing radius or covering radius of a code C. We consider in this thesis both definitions applied to binary codes.

In the next definition we consider the intrinsic capacity a code has to correct up to a given number of errors, it means it is independent from the effect of the channel. Therefore, it is not a probability.

Definition. 2.1.1 (Proportion of correct decoding)**.** *Given C a q-ary code of length n, $e \in \mathbb{N}$. We define the* ***proportion of correct decoding*** *up to e errors for the code C as*

$$\mathrm{PCD}_{\leqslant \mathrm{e}}(\mathrm{C}) := \frac{\displaystyle\sum_{\mathbf{c}\in C}\left(\sum_{t=0}^{e}\left|S_t(\boldsymbol{c}) \setminus \bigcup_{\boldsymbol{c}'\in C\setminus\{\boldsymbol{c}\}} B_t(\boldsymbol{c}')\right|\right)}{\displaystyle\sum_{\boldsymbol{c}\in C}|B_e(\boldsymbol{c})|}, \tag{2.2}$$

where the sphere of radius r and centered in $\boldsymbol{c}$ is denoted by

$$\mathrm{S}_r(\mathbf{c}) = \{\mathbf{x} \in \mathbb{F}_q^n : \mathrm{d}(\mathbf{x},\mathbf{c}) = r\}$$

and the ball centered in c of radius r is denoted by

$$\mathrm{B}_r(\mathbf{c}) = \{\mathbf{x} \in \mathbb{F}_q^n : \mathrm{d}(\mathbf{x},\mathbf{c}) \leqslant r\} = \cup_{i=0}^{r}\mathrm{S}_r(\mathbf{c}).$$

We note that, considering C to be a linear code, the quantities $|S_t(\mathbf{c}) \setminus \bigcup_{\mathbf{c}'\in C\setminus\{\mathbf{c}\}} B_t(\mathbf{c}')|$ and $|B_e(\mathbf{c})|$ do not depend on $\mathbf{c} \in C$, so we redefine (2.2) as follows

$$\mathrm{PCD}_{\leqslant \mathrm{e}}(\mathrm{C}) := \frac{\displaystyle\sum_{t=0}^{e}\left|S_t(\mathbf{0}) \setminus \bigcup_{\mathbf{0}\neq\mathbf{c}\in C} B_t(\mathbf{c})\right|}{|B_e(\mathbf{0})|}. \tag{2.3}$$

It is then clearer that this value does not depend on the codeword that is sent, but on the code to which it belongs.

As an illustration, we can see in the Figure 2 which are the codewords that are counted and which are not in $\mathrm{PCD}_{\leqslant \mathrm{e}}(\mathrm{C})$. Here we represent by x and o the codewords in $\mathrm{B}_r(\mathbf{0})$, where the x cannot be uniquely decoded and instead o can, at least with respect to the codeword c. The same is considered for any other $\mathbf{c} \in C \setminus \{\mathbf{0}\}$.

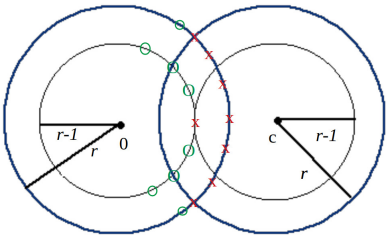


Figure 2 – Illustration of $\mathrm{PCD}_{\leqslant e}(\mathrm{C})$

If on the other hand, we need the capacity the code has to correct an exact number of errors, then we introduce:

Definition. 2.1.2 (Proportion of Exact Decoding)**.** *Given C a q-ary code of length n, $e \in \mathbb{N}$. We define the* ***proportion of exact decoding*** *e errors for the code C as*

$$\mathrm{PED}_{=\mathrm{e}}(\mathrm{C}) := \frac{\sum_{\mathbf{c}\in C}\left(|S_e(\boldsymbol{c}) \setminus \bigcup_{\mathbf{c}'\in C\setminus\{\mathbf{c}\}} B_e(\boldsymbol{c'})|\right)}{\sum_{\mathbf{c}\in C}(|S_e(\boldsymbol{c})|)}. \tag{2.4}$$

And as for Definition 2.1.1 we can for linear codes simply consider $\mathbf{c} = \mathbf{0}$ and then (2.4) turns into

$$\mathrm{PED}_{=\mathrm{e}}(\mathrm{C}) := \frac{|S_e(\mathbf{0}) \setminus \bigcup_{\mathbf{0}\neq\mathbf{c}\in C} B_e(\mathbf{c})|}{|S_e(\mathbf{0})|}. \tag{2.5}$$

Example. 2.1.3. *To illustrate these concepts presented in definitions 2.1.1 and 2.1.2 we consider two codes, for which these values were calculated. The $[7,4]_2$-Hamming Code $C_{H,7}$ (which is perfect) and one of the best known linear codes of length 11, obtained from the database in* Magma*, which we denote by $C_{[11,4,5]}$.*

$C = C_{H,7}$						
Weight enumerator				$x^7 + 7x^4 + 7x^3 + 1$		
k	1	2	3	4		
$\mathrm{d}_k(C)$	3	5	6	7		
e	1	2	3	4	5	$\cdots$
$\mathrm{PCD}_{\leqslant \mathrm{e}}(\mathrm{C})$	1	0.276	0.12	0.08	0.06	$\cdots$
$\mathrm{PED}_{=\mathrm{e}}(\mathrm{C})$	1	0	$\cdots$	$\cdots$	$\cdots$	0

$C = C_{[11,4,5]}$						
Weight enumerator			$x^8 + 2x^7 + 6x^6 + 6x^5 + 1$			
k	1	2	3	4		
$\mathrm{d}_k(C)$	5	8	10	11		
e	1	2	3	4	5	$\cdots$
$\mathrm{PCD}_{\leqslant \mathrm{e}}(\mathrm{C})$	1	1	0.379	0.156	0.085	$\cdots$
$\mathrm{PED}_{=\mathrm{e}}(\mathrm{C})$	1	1	0.1272	0	$\cdots$	0

Table 1 – Proportions of correction for different binary codes

In Table 1 one can notice, that even beyond the packing radius (2 for $C_{[11,4,5]}$), some errors can be properly corrected. That is the meaning of having $\mathrm{PED}_{=3}(\mathrm{C}) = 0.1272$*, that approximately* 12% *of the errors with weight* 3 *can be corrected. The fact that it does not happen for the Hamming Code, that is, no error can be corrected beyond the covering radius, gives us a hint for the role of the covering radius. As we shall see later in this chapter, no error can be corrected at or beyond that point and there is always an error of weight e that can be properly corrected for every e smaller than the covering radius.*

On the other hand, as we could expect, $PCD_{\leqslant e}(C)$ decreases with e and it is always greater than 0, since messages with no error are always properly decoded, and this means that $PCD_{\leqslant n}(C) \geqslant q^{\delta_1}/q^n \geqslant 1/q^n$, for an $[n,k]_q$-code.

Now we can work properly an example that is more significant than the naive toy example given in the beginning of this chapter. We shall consider two codes C_1, C_2 both with equal classic parameters $[n,k,d]_q$ and yet $\mathrm{PCD}_{\leqslant e}(\mathrm{C}_1) \neq \mathrm{PCD}_{\leqslant e}(\mathrm{C}_2)$, for $e > \delta_1$. As we already mentioned, it is clear that if $e \leqslant \delta_1$, then $\mathrm{PCD}_{\leqslant \mathrm{e}}(\mathrm{C}) = 1$.

Example. 2.1.4. *In (PLESS, 1972a) Pless classified self-orthogonal code of length up to $n = 20$. Among those codes, we find the $[16,8]_2$ codes E_{16} and F_{16}, which are the only non-equivalent self-orthogonal (in fact self-dual) codes of length 16. They were chosen in order to compare their PCD and PED in the fairest conditions, since they have the same*

minimal distance. Moreover, we calculated, and they happen to have the same weight hierarchy, that is, $d_i(E_{16}) = d_i(F_{16})$ *for every* $1 \leqslant i \leqslant 8$*, as we can see in the second row of Table 2. However, they have different weight enumerators,* $x^{16}+28x^{12}+198x^8+28x^4+1$ *and* $x^{16} + 12x^{12} + 64x^{10} + 102x^8 + 64x^6 + 12x^4 + 1$*, respectively. This difference in the weight enumerator explains why* F_{16} *is expected to perform better:* F_{16} *has 12 codewords of weight* 4 *while* E_{16} *has* 28*.*

We calculated the invariants of both E_{16} *and* F_{16} *and they are summarized in Table 2.*

k or e	1	2	3	4	5	6	7	8
$\mathrm{d}_k(E_{16}) = \mathrm{d}_k(F_{16})$	4	6	8	10	12	14	15	16
$\mathrm{PCD}_{\leqslant \mathrm{e}}(\mathrm{E}_{16})$	1	0.124	0.0244	0.0067	0.0025	$\cdots$		
$\mathrm{PCD}_{\leqslant \mathrm{e}}(\mathrm{F}_{16})$	1	0.59	0.116	0.032	0.0117	$\cdots$		
$\mathrm{PED}_{=\mathrm{e}}(\mathrm{E}_{16})$	1	0	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	0
$\mathrm{PED}_{=\mathrm{e}}(\mathrm{F}_{16})$	1	0.53	0	$\cdots$	$\cdots$	$\cdots$	$\cdots$	0

Table 2 – Comparison of E_{16} and F_{16} error correction capabilities

In Table 2 it can be seen that not only both codes E_{16} and F_{16} have the same 1st-minimum distance, but the hierarchical weight distribution is also the same for them. Up to this point, we could expect that they will perform in the same way in what concerns $\mathrm{PCD}_{\leqslant e}$ and $\mathrm{PED}_{=e}$. However, we explicitly calculated the number of subcodes of dimension j, for $j \leqslant 4$ that reach the distance $d_j(C)$[1]. They are represented by the arrays $(28; 56; 70; 56)$ for E_{16} and $(12; 8; 2; 8)$ for F_{16}. Having F_{16} less subspaces with minimum support for each dimension $j \in \{1, 2, 3, 4\}$, it allows the received vectors to be covered by fewer codewords and this increases the values of both $\mathrm{PCD}_{\leqslant e}(\mathrm{C})$ and $\mathrm{PED}_{=e}(\mathrm{C})$, as seen in Table 2.

As we mentioned in the beginning of this section, it is well known that every codeword received with less errors than δ_1 are always corrected. In consequence, $\mathrm{PED}_{=e}(C) = 1$ for $e \leqslant R_{pack}(C)$. In the following remark we notice what is expected but unknown: $\mathrm{PED}_{=e}(C) = 0$, for $e > R_{cov}(C)$.

Remark 2.1.1. *Let* $C \subseteq \mathbb{F}_q^n$*, with covering radius* $r := R_{cov}(C)$*. That* $\mathrm{PED}_{=\mathrm{e}}(\mathrm{C}) = 0$*, for* $e > r$ *is a direct consequence of the definition of covering radius, since every* $\mathbf{v} \in S_{r+1}(\boldsymbol{c})$

[1] Due to the size of these codes, only up to dimension 4 it was possible to determine the number of subspaces that had the given minimum support.

belongs to a ball centered at another codeword $\mathbf{c}' \in C \setminus \{\mathbf{c}\}$.

In the particular case of perfect codes, where $R_{pack}(C) = R_{cov}(C)$, we have the following.

Proposition 2.1.2. *If* $C \subseteq \mathbb{F}_q^n$ *is a perfect code, then* $\mathrm{PED}_{=\mathrm{e}}(\mathrm{C}) = 0$, *for* $e > R_{pack}(C)$, *its packing radius.*

Remark 2.1.5. *The converse is not true in general, i.e., that* $\mathrm{PED}_{=\mathrm{e}}(\mathrm{C}) = 0$, *for* $e > R_{pack}(C)$ *does not imply that* C *is a perfect code. For instance, if we consider* $C \subseteq \mathbb{F}_2^6$ *with generator matrix*

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

We can do by extension the sets $B_1(\boldsymbol{c}), \boldsymbol{c} \in C$ *and join them to finally compute that* $|\bigcup_{\mathbf{c}\in C} B_1(\boldsymbol{c})| = 56$. C *has covering radius 2, then* $|\bigcup_{\mathbf{c}\in C} B_2(\boldsymbol{c})| = 64$. *Therefore,* C *is not perfect and yet* $\mathrm{PED}_{=\mathrm{e}}(\mathrm{C}) = 0$, *for* $e > 1$. *In fact, we can see the different balls of radius 1 centered in codewords of* C *as follows:*

000000	100101	110011	010110	011001	111100	101010	001111
000001	100100	110010	010111	011000	111101	101011	001110
001000	101101	111011	011110	010001	110100	100010	000111
000100	100001	110111	010010	011101	111000	101110	001011
010000	110101	100011	000110	001001	101100	111010	011111
100000	000101	010011	110110	111001	011100	001010	101111
000010	100111	110001	010100	011011	111110	101000	001101

Table 3 – Set of all $B_1(\mathbf{c})$, $\mathbf{c} \in C$.

In Table 3 the red row contains the codewords and below each one the 6 vectors at distance 1. Here we can check that 110000, 001100 *are not elements of such balls, meaning the whole ambient space has not been covered by them.*

$B_2(001111)$	$B_2(111100)$	$B_2(110011)$
0 0 1 0 0 1	1 1 1 0 1 0	1 1 0 1 0 1
0 0 0 1 1 0	1 1 0 1 0 1	1 1 1 0 1 0
1 0 0 1 1 1	**0 1 0 1 0 0**	0 1 1 0 1 1
0 0 1 1 0 0	1 1 1 1 1 1	**1 1 0 0 0 0**
1 0 1 1 0 1	0 1 1 1 1 0	**0 1 0 0 0 1**
0 0 0 1 1 1	1 1 0 1 0 0	1 1 1 0 1 1
0 1 1 0 1 1	**1 0 1 0 0 0**	1 0 0 1 1 1
0 0 1 1 0 1	1 1 1 1 1 0	1 1 0 0 0 1
0 1 1 1 1 0	1 0 1 1 0 1	**1 0 0 0 1 0**
1 1 1 1 1 1	**0 0 1 1 0 0**	**0 0 0 0 1 1**
0 0 1 0 1 0	1 1 1 0 0 1	1 1 0 1 1 0
1 0 1 0 1 1	**0 1 1 0 0 0**	0 1 0 1 1 1
1 0 1 1 1 0	0 1 1 1 0 1	**0 1 0 0 1 0**
0 1 1 1 1 1	1 0 1 1 0 0	1 0 0 0 1 1
0 0 1 0 1 1	1 1 1 0 0 0	1 1 0 1 1 1
0 0 0 1 0 1	1 1 0 1 1 0	1 1 1 0 0 1
0 0 1 1 1 0	1 1 1 1 0 1	1 1 0 0 1 0
1 0 1 1 1 1	0 1 1 1 0 0	0 1 0 0 1 1
0 0 1 1 1 1	1 1 1 1 0 0	1 1 0 0 1 1
0 1 0 1 1 1	**1 0 0 1 0 0**	1 0 1 0 1 1
0 0 0 0 1 1	**1 1 0 0 0 0**	1 1 1 1 1 1
0 1 1 1 0 1	1 0 1 1 1 0	**1 0 0 0 0 1**

Table 4 – Set of balls of radius two centered at three codewords of C that cover $S_2(\mathbf{0})$.

On the other hand, to check that effectively $\mathrm{PED}_{=\mathrm{e}}(\mathrm{C}) = 0$, *for* $e > 1$, *consider when* $e = 2$. *Here we consider first by the definition of* $\mathrm{PED}_{=\mathrm{e}}(\mathrm{C})$ *the set*

$$S_2(\mathbf{0}) := \{(000110), (001001), (101000), (000011), (100010), (010100), (010001),$$
$$(110000), (000101), (100100), (100001), (001010), (011000), (001100), (010010)\}$$

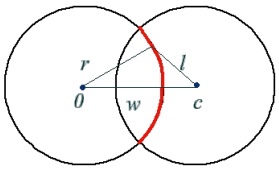
and check that it is covered by the balls $B_2((001111)), B_2((101010)), B_2((110011))$, *see Table 2.1.5. The 15 elements of* $S_2(\mathbf{0})$ *appear in boldface, the colored ones are repeated.*

2.2 Bounds for $\mathrm{PCD}_{\leqslant e}(C)$

Calculating the proportion of correct decoding for a code is a hard problem, so we shall look for bounds. One first approach to obtain bounds for $\mathrm{PCD}_{\leqslant e}(\mathrm{C})$ is to find an expression for $\mathrm{S}_r(\mathbf{0}) \cap \mathrm{B}_r(\mathbf{c})$. For that it is necessary to consider what vectors are in this set, i.e.,

$$S_r(\mathbf{0}) \cap B_r(\mathbf{c}) := \left\{\mathbf{v} \in \mathbb{F}_q^n \mid \mathrm{wt}(\mathbf{v}) = r \text{ and } \mathrm{d}(\mathbf{v}, \mathbf{c}) \leqslant r\right\}.$$

Then as can be seen in the next figure we exclude only the part of the sphere that is common with any other ball of radius r centered in codewords.



To determine the intersection of balls excluding their centers, we consider $r < d(\mathbf{0}, \mathbf{c}) = \mathrm{wt}(\mathbf{c})$. We denote $w := \mathrm{wt}(\mathbf{c})$. At the same time, the closest a vector in the intersection can be to $\mathbf{c}$ is $w - r$, then

$$w - r \leqslant \mathrm{d}(\mathbf{v}, \mathbf{c}) \leqslant r.$$

Let us define

$$i := |\{s : v_s = 1 \text{ and } c_s = 1\}|.$$

So if i coordinates of $\mathbf{v}$ belong to $\mathrm{Supp}(\mathbf{c})$, then $r - i$ are non-zero coordinates among the $n - w$ other possible options. This means that for every choice of i ones contained in the support of $\mathbf{c}$, there are $r - i$ options in its complement. Now we need to guarantee that $\mathrm{d}(\mathbf{v}, \mathbf{c})$ belongs to the closed interval $[w - r, r]$, then $((w - i) + (r - i)) \in [w - r, r]$, or

$$w - r \leqslant (w - i) + (r - i) \leqslant r$$

$$\Rightarrow w - r \leqslant (w + r - 2i) \leqslant r$$

$$\Rightarrow w - 2r \leqslant w - 2i \leqslant 0$$

$$\Rightarrow -2r \leqslant -2i \leqslant -w$$

$$\Rightarrow r \geqslant i \geqslant \frac{w}{2}$$

Remark. We can just consider $\lceil \frac{w}{2} \rceil$, in general, just in case w is odd and still the inequation holds. So we get that for i coordinates taken from the w possible in the $\mathrm{Supp}(\mathbf{c})$ we take $r - s$ among the $n - w$ others, for $i \in [\lceil \frac{w}{2} \rceil, r]$. Thus

$$|S_r(\mathbf{0}) \cap B_r(\mathbf{c})| = \sum_{i=\lceil \frac{w}{2} \rceil}^{r} \binom{w}{i} \binom{n-w}{r-i}. \tag{2.6}$$

Example. 2.2.1. *We compute for a code with parameters* $[16, 8, 4]_2$*, considering a random codeword* $\mathbf{c}$ *with* $\mathrm{wt}(\boldsymbol{c}) = 8$ *and for* $r = 5$*. Then*

$$|S_5(\mathbf{0}) \cap B_5(\boldsymbol{c})| = \sum_{i=\lceil \frac{8}{2} \rceil}^{5} \binom{8}{i} \binom{16-8}{5-i} = \binom{8}{5}\binom{8}{1} + \binom{8}{5}\binom{8}{0} = 70 \cdot 8 + 56 \cdot 1 = 616.$$

2.2.0.1 INTERSECTION OF TWO BALLS

Now we can use Equation 2.6 to get the intersection of two balls of radius r. In general, for every vector of weight $j \in [w-r, r]$, where $w \geqslant r$, if we consider as previously that i 1's of $\mathbf{v}$ lay on $\mathrm{Supp}(\mathbf{c})$, then $j-i$ lay among the $n-w$ other ones. Here

$$\mathrm{d}(\mathbf{v}, \mathbf{c}) \in [w-j, r] \Rightarrow (w-i) + (j-i) \in [w-j, r].$$

Then

$$w - j \leqslant w - i + j - i \leqslant r$$

so that

$$-2j \leqslant -2i \leqslant r - w - j$$

and

$$j \geqslant i \geqslant \frac{w + j - r}{2}$$

This means that we have the following proposition.

Proposition 2.2.1. *Given a codeword* $\mathbf{c}$ *of weight* w*, the number of vectors in the intersection of the balls of radius* r *centered at* $\boldsymbol{c}$ *and* $\mathbf{0}$ *is*

$$|B_r(\mathbf{0}) \cap B_r(\boldsymbol{c})| = \sum_{j=w-r}^{r} \sum_{i=\lceil \frac{w+j-r}{2} \rceil}^{j} \binom{w}{i} \binom{n-w}{j-i}. \tag{2.7}$$

This way, we can count the vectors in the shadow zone in the next figure.

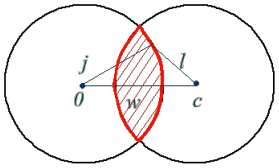


Figure 3 – Intersection of balls with same radius centered at $\mathbf{0}$ and $\mathbf{c}$, $\mathrm{wt}(\mathbf{c}) = w$

Example. 2.2.2. *Under the same conditions as for Example 2.2.1 we have by means of Proposition 2.2.1 that:*

$$|B_5(\mathbf{0}) \cap B_5(\boldsymbol{c})| = \sum_{j=3}^{5} \sum_{i=\lceil \frac{3+j}{2} \rceil}^{j} \binom{8}{i} \binom{8}{j-i}$$

$$\Rightarrow |B_5(\mathbf{0}) \cap B_5(\boldsymbol{c})| = \binom{8}{3}\binom{8}{0} + \binom{8}{4}\binom{8}{0} + \binom{8}{4}\binom{8}{1} + \binom{8}{5}\binom{8}{0} = 742.$$

Considering the weight of the codewords in C we introduce this definition that allows us to distinguish the codewords of minimum weight in particular.

Definition. 2.2.3 (i-th subset of a code C)**.** *Given a code C we define the i-**th subset of** C as all the codewords of the code with weight i, i.e.,*

$$C_i := \{\mathbf{c} \in C : |\mathrm{Supp}(\boldsymbol{c})| = i\}.$$

Then we can immediately see that the code can be decomposed as

$$C := \dot{\bigcup_{j=0}^{n}} C_j.$$

Next we define formally what we call the spectrum of a code C.

Definition. 2.2.4 (Spectrum)**.** *Given $C \leqslant \mathbb{F}_q^n$ we define the **spectrum** of C denoted by $Spec(C)$ as the matrix $(A_i^j)_{k\times n}$, where*

$$A_i^j := |\{D \subset C : \dim(D) = i \text{ and } ||D|| := |\mathrm{Supp}(D)| = j\}|.$$

If $i = 1$, then $\sum_{j=1}^{n} A_1^j x^j$ is also the polynomial weight enumerator of C.

In the following remark, we take a first step into finding an appropriate bound for $\mathrm{PCD}_{\leqslant e}(C)$, for $e < \mathrm{d}(C)$.

Remark 2.2.2. *For $C \leqslant \mathbb{F}_q^n$, one can verify that:*

a.) If $d_2(C) \geqslant 2d_1(C)$, then $A_1^{d_1(C)} \leqslant \dfrac{n}{d_1(C)}$. This is, the number of one-dimensional subspaces that reach the minimum support is bounded by how many times the 1st minimum weight fits in the length of the code, whenever the 2nd minimum distance is at least twice the first one.

b.) If $\{\mathbf{c}_1, \ldots, \mathbf{c}_m\}$ is the set of codewords with minimum weight in C and all the subspaces that reach the $i-th$ minimum distance are contained in $\langle \mathbf{c}_1, \ldots, \mathbf{c}_m \rangle$, for every i. Then, $A_1^{d_1(C)} = m$ and $A_i^{d_i(C)} \leqslant \binom{m}{i}$, for $i \leqslant k = \dim(C)$, with $D := \langle \mathbf{c}_{h_1}, \ldots, \mathbf{c}_{h_i} \rangle \subset C$, such that $\dim(D) = i$ and $||D|| = d_i(C)$. This is, if every

subspace that is counted in $A_i^{d_i(C)}$ is spanned by the codewords of minimum weight in C, then $A_i^{d_i(C)} \leqslant \binom{m}{i}$. In particular, the first one would be m, the number of such codewords ($A_1^{d_1(C)} = m$).

c.) If $q = 2$, given any two codewords of C, we can verify that

$$\begin{aligned} \mathrm{wt}(\mathbf{c}_1) + \mathrm{wt}(\mathbf{c}_2) &= \mathrm{wt}(\mathbf{c}_1 + \mathbf{c}_2) + 2|\mathrm{Supp}(\mathbf{c}_1) \cap \mathrm{Supp}(\mathbf{c}_2)| \\ \Rightarrow |\mathrm{Supp}(\mathbf{c}_1) \cap \mathrm{Supp}(\mathbf{c}_2)| &= \frac{\mathrm{wt}(\mathbf{c}_1) + \mathrm{wt}(\mathbf{c}_2) - \mathrm{wt}(\mathbf{c}_1 + \mathbf{c}_2)}{2}. \\ |\mathrm{Supp}(\mathbf{c}_1) \setminus \mathrm{Supp}(\mathbf{c}_2)| + |\mathrm{Supp}(\mathbf{c}_2) \setminus \mathrm{Supp}(\mathbf{c}_1)| &= d(\mathbf{c}_1, \mathbf{c}_2) \geqslant d_1(C). \end{aligned}$$

2.2.0.2 INCLUSION-EXCLUSION PRINCIPLE

We have been trying to determine and use the number of elements in the intersection of pairs and triplets of balls with the same radius centered in codewords, because due to the definition of the Proportions $\mathrm{PCD}_{\leqslant e}(C)$ and $\mathrm{PED}_{=e}(C)$, they depend on the intersection that the union of all balls centered at non-zero codewords has with the layered ball of radius r and centered at the null-vector. This implies that we need to use the inclusion-exclusion principle, which in the simplest cases states that if we have M and N finite sets. The size of their union can be calculated using

$$|M \cup N| = |M| + |N| - |M \cap N|$$

this, since when computing $|M| + |N|$ we count twice the elements of $M \cap N$. This redundancy is then compensated by subtracting $|M \cap N|$.

Now let us consider the scenario in which we have three finite sets, let us say L, M and N. A counting exercise will show that

$$|L \cup M \cup N| = |L| + |M| + |N| - |L \cap M| - |L \cap N| - |M \cap N| + |L \cap M \cap N|. \quad (2.8)$$

We could simply consider that the size of the union is $|L| + |M| + |N|$ but this is again wrong because the elements in $L \cap M$, $L \cap N$, and $M \cap N$ would have been counted too many times. Therefore, by subtracting $|L \cap M| + |L \cap N| + |M \cap N|$, we try to eliminate this over-counting but then notice that $L \cap M \cap N$ has been excluded more than appropriate. And then we compensate by adding once $|L \cap M \cap N|$.

Example. 2.2.5. *Let* $A = \{1,2,3,4\}$, $B = \{3,4,5,6\}$ *and* $C = \{2,3,4,5,6,7\}$. *Now Equation (2.8) says* $7 = 4+4+6-2-3-4+2$, *which is happily true.*

In general, we have:

Theorem 2.2.3 (Inclusion-Exclusion principle)**.** *Given* $n \in \mathbb{N}$ *and finite sets* A_i *for* $1 \leqslant i \leqslant n$. *Then*

$$\left|\bigcup_{1\leqslant i\leqslant n} A_i\right| = \sum_{1\leqslant i_1\leqslant n} |A_{i_1}| - \sum_{1\leqslant i_1\leqslant i_2\leqslant n} |A_{i_1}\cap A_{i_2}| + \sum_{1\leqslant i_1\leqslant i_2\leqslant i_3\leqslant n} |A_{i_1}\cap A_{i_2}\cap A_{i_3}| - \ldots + (-1)^{n+1}\left|\bigcap_{i=1}^{n} A_i\right|.$$

Its proof is natural from counting the elements as can be seen in (BONFERRONI, 1936).

It is also necessary to check that by considering the higher intersections the expression to the right in Theorem 2.2.3 becomes closer to the union than considering lesser terms. And it is what the following remark deals with.

Remark 2.2.4. *We can notice that every term to the right is smaller than the ones to the left, by being calculated through computing the intersection with each time more sets, and also having the alternating signs leads to an oscillating value towards the exact value of the union and as when reaching the last term the value is exactly the number of elements in the union of n sets, this means that the oscillation is asymptotic to this number, meaning that with each step the distance with respect to the final value gets smaller. Then we can say that for* $k \in \{1,\ldots,n\}$

$$\left|\left|\bigcup_{1\leqslant i\leqslant n} A_i\right| - \sum_{j=1}^{k-1}(-1)^{j+1}\left(\sum_{1\leqslant i_1<\cdots<i_j\leqslant n} |A_{i_1}\cap\ldots\cap A_{i_j}|\right)\right|$$

$$\geqslant \left|\left|\bigcup_{1\leqslant i\leqslant n} A_i\right| - \sum_{j=1}^{k}(-1)^{j+1}\left(\sum_{1\leqslant i_1<\cdots<i_j\leqslant n} |A_{i_1}\cap\ldots\cap A_{i_j}|\right)\right|$$

From Remark 2.2.2 we get some bounds for some coefficients of $Spec(C)$. And making use of the first step of Theorem 2.2.3, this is considering only the exclussion of the intersection with the individual sets, one gets the following proposition.

Proposition 2.2.5. *Given a code* C, $e \in \mathbb{N}$. *Then*

$$\mathrm{PCD}_{\leqslant e}(C) \geqslant \frac{\sum_{t=0}^{\lfloor \frac{w-1}{2} \rfloor} |S_t(\mathbf{0})| + \sum_{t=\lfloor \frac{w}{2} \rfloor}^{e} \left(|S_t(\mathbf{0})| - \sum_{j=1}^{2t} |C_j||S_t(\mathbf{0}) \cap B_t(\mathbf{c}_j)| \right)}{|B_e(\boldsymbol{c})|},$$

where $\mathbf{c}_j \in C_j$ *and* $w = \mathrm{d}_1(C)$*. Moreover, if* C *is a constant weight linear code, then*

$$\mathrm{PCD}_{\leqslant e}(C) \geqslant \frac{\sum_{t=0}^{\lfloor \frac{w-1}{2} \rfloor} \binom{n}{t} + \sum_{t=\lfloor \frac{w}{2} \rfloor}^{e} \left(\binom{n}{t} - |C_w| \sum_{i=\lceil \frac{w}{2} \rceil}^{t} \binom{w}{i} \binom{n-w}{t-i} \right)}{\sum_{i=0}^{e} \binom{n}{i}}.$$

In order to apply the inclusion-exclusion principle to get bounds for $\mathrm{PCD}_{\leqslant e}(C)$ and $\mathrm{PED}_{=e}(C)$, respectively, we need to introduce some definitions that will make the notation lighter. Also, whenever necessary, given a set $X \subseteq [k]$, we will denote by $c_X = \{\mathbf{c_i} | i \in X\}$, the codewords indexed by X. So we have for example $\{\mathbf{c_1}, \ldots, \mathbf{c_{2h}}\} = c_{[2h]}$.

Definition. 2.2.6. *Given* $\mathbf{c_1}, \mathbf{c_2} \in C, t \in \mathbb{N}$*. We define the* ***set of triplets t-intersection*** *of* C *as:*

$$T_t^3 := \left\{ \{\mathbf{c}_1, \mathbf{c}_2\} \middle| \{\mathbf{c}_1, \mathbf{c}_2\} \text{ is l.i. and } \Xi^t_{\{\mathbf{c}_1, \mathbf{c}_2\}} \neq \varnothing \right\},$$

where $\Xi^t_{\{\mathbf{c}_1, \mathbf{c}_2\}} := B_t(\mathbf{c}_1) \cap B_t(\mathbf{c}_2) \cap S_t(\mathbf{0})$*. A* ***minimal set of triplets t-intersection*** $\mathcal{T}_t^3 \subset T_t^3$ *of* C *is one of the smallest subsets such that*

$$\bigcup_{c_{[2]} \in T_t^3} \Xi^t_{c_{[2]}} = \bigcup_{c_{[2]} \in \mathcal{T}_t^3} \Xi^t_{c_{[2]}}$$

Then using this notation we get the following bounds, which are just a restatement of the inclusion-exclusion principle.

Proposition 2.2.6.

$$\frac{\sum_{t=0}^{\lfloor \frac{w-1}{2} \rfloor} |S_t(\mathbf{0})| + \sum_{t=\lfloor \frac{w}{2} \rfloor}^{e} \left(|S_t(\mathbf{0})| - \sum_{j=1}^{2t} |C_j||S_t(\mathbf{0}) \cap B_t(c_j)| + \sum_{c_{[2]} \in \mathcal{T}_t^3} \left| \Xi^t_{c_{[2]}} \right| \right)}{|B_e(\boldsymbol{c})|}$$

$$\geqslant \mathrm{PCD}_{\leqslant e}(C) \geqslant \frac{\sum_{t=0}^{\lfloor \frac{w-1}{2} \rfloor} |S_t(\mathbf{0})| + \sum_{t=\lfloor \frac{w}{2} \rfloor}^{e} \left(|S_t(\mathbf{0})| - \sum_{j=1}^{2t} |C_j||S_t(\mathbf{0}) \cap B_t(\mathbf{c}_j)| \right)}{|B_e(\boldsymbol{c})|},$$

where $\mathbf{c}_j \in C_j$ *and* $w = \mathrm{d}_1(C)$*.*

Proof. From Definition 2.1.1, Equation 2.3, what we need to get to these bounds is to sequentially obtaining

$$\left| S_t(\mathbf{0}) \setminus \bigcup_{\mathbf{0} \neq \mathbf{c} \in C} B_t(\mathbf{c}) \right|.$$

We already know that if $t \leqslant \delta_1$ it turns out to be only $S_t(\mathbf{0})$. The rest of the expression is a result from applying Theorem 2.2.3 and Remark 2.2.4. By doing this, if $\left\lfloor \frac{w-1}{2} \right\rfloor \leqslant t \leqslant e, w = d_1(C)$, as seen in Proposition 2.2.5 we get a lower bound of $\mathrm{PCD}_{\leqslant e}(C)$. This because out of the elements of $S_t(\mathbf{0})$ we remove in a first step every non-empty intersection it has with balls of radius t centered at codewords. To get the upper bound, now by Theorem 2.2.3 we add now the non-empty intersections of $S_t(\mathbf{0})$ with two balls of radius t centered at codewords.

□

Unless we know for sure that any of the intersections with a higher number of balls of radius t centered at codewords are empty, these expressions remain simply a bound.

Remark 2.2.7. *With this perspective, it would be interesting to find some co-relation between a code having smaller generalized Hamming weight $d_k(C)$, compared to another one, and the numbers $A_k^{d_k(C)}$. That is, being able to say something about the quantity of k-dimensional subspaces that meet the minimum support. But even though according to the observations seen on the spectrum of codes with length smaller or equal to* 11*, there seems to be a tendency that for C_1, C_2, two comparable codes and $k \leqslant \dim(C_1) = \dim(C_2)$,*

$$d_k(C_1) < d_k(C_2) \Rightarrow A_k^{d_k(C_1)} < A_k^{d_k(C_2)}, \tag{2.9}$$

which I must admit is even counter-intuitive. But it is also in general not true. In fact, if we consider

$$G_1 := \begin{pmatrix} 1000111111 \\ 0100111001 \\ 0010011110 \\ 0001001111 \end{pmatrix} G_2 := \begin{pmatrix} 1000101000 \\ 0100010001 \\ 0010110000 \\ 0001000110 \end{pmatrix},$$

*then we have two [10,4]-codes, where the one generated by G_1 has a generalized weight distribution of [4,**6**,8,10] and for each dimension the number of subspaces that meet that size of support is (4,**1**,1,1) and the one generated by G_2 has a generalized weight distribution of [3,**5**,7,10] and for each dimension the number of subspaces that meet that size of support is (4,**2**,1,1). Here we notice that even when $d_2(C_1) = 6 > d_2(C_2) = 5$ we see that $A_2^{d_2(C_1)} = 1 < 2 = A_2^{d_2(C_2)}$. This seems to be more natural than Equation 2.9, but not in*

general true. This tells us that, before hand, we cannot say something on what to expect in relation to the spectrum. This based only in the weight hierarchy of the codes, without analyzing the particular case.

We have seen that the weight hierarchy gives some hints on the coefficients of $Spec(C)$ and also how to determine the bounds we are looking for. So to continue this path we now generalize Definition 2.2.6 for k-tuples as follows:

Definition. 2.2.7. *Given* $\mathbf{c_1}, \ldots, \mathbf{c_{k-1}}$ *in* $C, t \in \mathbb{N}$. *We define the* ***set of*** k***-tuples*** t***-intersection*** *of* C *as:*

$$T_t^k := \left\{ c_{[k-1]} \middle| c_{[k-1]} \text{ is l.i. and } \Xi^t_{c_{[k-1]}} \neq \varnothing \right\},$$

where $\Xi^t_{c_{[k-1]}} := B_t(\mathbf{c_1}) \cap \cdots \cap B_t(\mathbf{c_{k-1}}) \cap S_t(\mathbf{0})$. *A* ***minimal set of*** k***-tuples*** t***-intersection*** $\mathcal{T}_t^k \subset T_t^k$ *of* C *is one of the smallest subsets such that*

$$\bigcup_{c_{[k-1]} \in T_t^k} \Xi^t_{c_{[k-1]}} = \bigcup_{c_{[k-1]} \in \mathcal{T}_t^k} \Xi^t_{c_{[k-1]}}$$

Remark 2.2.4 guarantees that these bounds are tighter with each step. The next proposition tell us that if C is such that it meets the condition of Remark 2.2.2, then we get additional bounds for the coefficients of $Spec(C)$.

Proposition 2.2.8. *If we have that* C *has the property that the subcodes of dimension* k *that have minimum support are spanned by words of minimum weight, then* $|\mathcal{T}_t^{k+1}| \leqslant A_k^{d_k(C)}, t \geqslant \left\lceil \frac{w}{2} \right\rceil$. *Furthermore, if those are the only codewords with weight smaller or equal to* $2t$, *then* $|\mathcal{T}_t^{k+1}| \leqslant \binom{m}{k}$. *In particular, for* $k = 2$ *we have ,* $|\mathcal{T}_t^3| \leqslant A_2^{d_2(C)} \leqslant \binom{m}{2}$.

Proof. The fact that $|\mathcal{T}_t^{k+1}| \leqslant A_k^{d_k(C)}, t \geqslant \left\lceil \frac{w}{2} \right\rceil$, is a consequence of the definition of $\mathcal{T}_t^{k+1}$. Because, by definition, it is the smallest set of linearly independent k-tuples of C that additionally are centers of balls not disjoint to $S_t(\mathbf{0})$. Under the assumption that the subspaces that are counted in $A_k^{d_k(C)}$ are generated by codewords with minimum weight, then the k-tuples that span them contain $\mathcal{T}_t^{k+1}$. The upper bound is a result from counting all the possibilities as remarked in 2.2.2 b). □

This property mentioned in the previous proposition was observed in the binary Hamming code of length 7. But it was not found as a general condition for every code.

To structure the following expressions, let's consider

$$\Delta := \sum_{t=0}^{\lfloor \frac{w-1}{2} \rfloor} |S_t(\mathbf{0})| + \sum_{t=\lfloor \frac{w}{2} \rfloor}^{e} \left(|S_t(\mathbf{0})| - \sum_{j=1}^{2t} |C_j||S_t(\mathbf{0}) \cap B_t(\mathbf{c}_j)| \right),$$

that is constant for a given $e \geqslant \lfloor \frac{w}{2} \rfloor$.

Then, we can generalize the previous bounds given in Proposition 2.2.6, considering again all the terms of the inclusion-exclusion principle of Theorem 2.2.3 and the fact that with each step the expression get closer to the exact value, as indicated in the Remark 2.2.4. This results in

Theorem 2.2.9. *Given $l \leqslant \lfloor \frac{k}{2} \rfloor$, where $k := \dim(C)$, we have:*

$$\frac{\left(\Delta + \sum_{t=\lfloor \frac{w}{2} \rfloor}^{e} \sum_{h=1}^{l} \sum_{c_{[2h]} \in \mathcal{T}_t^{2h+1}} \left| \Xi^t_{c_{[2h]}} \right| - \sum_{t=\lfloor \frac{w}{2} \rfloor}^{e} \sum_{h=2}^{l} \sum_{c_{[2h-1]} \in \mathcal{T}_t^{2h}} \left| \Xi^t_{c_{[2h-1]}} \right| \right)}{|B_e(\boldsymbol{c})|} \geqslant PCD_{\leqslant e}(C)$$

$$\geqslant \frac{\left(\Delta + \sum_{t=\lfloor \frac{w}{2} \rfloor}^{e} \sum_{h=1}^{l-1} \sum_{c_{[2h]} \in \mathcal{T}_t^{2h+1}} \left| \Xi^t_{c_{[2h]}} \right| - \sum_{t=\lfloor \frac{w}{2} \rfloor}^{e} \sum_{h=2}^{l-1} \sum_{c_{[2h-1]} \in \mathcal{T}_t^{2h}} \left| \Xi^t_{c_{[2h-1]}} \right| \right)}{|B_e(\boldsymbol{c})|}$$

Just as in Proposition 2.2.6 the difference between the upper and lower bound depends on the number of elements considered between the intersection of $2l$ balls and $2(l-1)$ with the sphere. Then due to Remark 2.2.4 one knows this gap is smaller with each step, and in the last step there will be none, meaning it will be the exact value of $\mathrm{PCD}_{\leqslant e}(C)$.

2.3 Closed expressions for perfect codes

Given these bounds we would like to determine how many steps are necessary to get to the exact value. We know that it is always reached by means of the inclusion-exclusion principle. But we also wish to relate them to the code's weight distribution, which provides further information than its minimum distance. It is this that will eventually allow us to distinguish between codes with equal minimum distance but different weight spectrum, resulting in different correction capabilities. In this sense, we have two important remarks. But first let us see an example that illustrates a situation for the first statement, on the number of necessary steps to reach the exact value.

Example. 2.3.1. *If we consider the code $C = C_{H,7}$ being the Hamming Code with parameters $[7,4,3]_2$, then we know for $e = 1$ the number of errors either $\mathrm{PCD}_{\leqslant e}(C)$ or $\mathrm{PED}_{=e}(C)$ are 1, because $e \leqslant \delta_1$. Now for $e = 2$ we get that the balls of radius two centered at the seven words of minimum weight intersected with the zero centered sphere are disjoint by pairs. And they represent the first step, then we expect and so it is that the exact value is already reached by the first step as follows: Let's consider as before C_j the set of the codewords of weight j. Take any $\mathbf{c_3} \in C_3, \mathbf{c_4} \in C_4$. Then we can obtain that for $e = 2$*

$$\mid S_2(\mathbf{0}) \cap B_2(\mathbf{c_3}) \mid = 3 \text{ and } \mid S_2(\mathbf{0}) \cap B_2(\mathbf{c_4}) \mid = 6.$$

That means there are intersections with the balls centered at words of weight 3 and 4, being even the intersection with codewords of weight 4 larger, but we noticed that in spite of that, the intersection with the codewords of weight 3 are disjoint and cover all the sphere meaning we get the exact amount $\mathrm{PCD}_{\leqslant e}(C)$ stands for. In fact

$$PCD_{\leqslant 2}(C) = 0,27586 = \frac{|B_1(\mathbf{0})| + |S_2(\mathbf{0})| - |C_3||S_2(\mathbf{0}) \cap B_2(\mathbf{c_3})|}{|B_2(\mathbf{0})|}$$
$$= \frac{8 + (21 - 7 \cdot 3)}{29} = 8/29.$$

Remark 2.3.1.

- *The number of necessary steps of the inclusion-exclusion principle to be applied depends on up to what point the Minimal set of k-tuples t- intersections are disjoint, that is $\mathcal{T}_t^k \subset T_t^k$ such that*

$$\bigcup_{c_{[k-1]} \in T_t^k} \Xi^t_{c_{[k-1]}} = \bigcup_{c_{[k-1]} \in \mathcal{T}_t^k} \Xi^t_{c_{[k-1]}}$$

have no intersection between them, being this the $(k-1)$-th step of the process and meaning the next l-tuples t-intersection will be empty and therefore do not count for the union considered, for $l \geqslant k$. And this leads us to the second remark.

- *If a code C is such that all the k-dimensional subcodes that reach the minimum support, that is $||D|| = \mathrm{d}_k(C)$, are generated by codewords of minimum weight and conversely every l.i. k set of codewords of minimum weight span a k-dimensional subcode with minimum support, then we have a direct influence of the spectrum (A_i^j) into the bounds, since then $|\mathcal{T}_t^{k+1}| = \binom{m}{k}$, where m is the number of codewords of minimum weight. Just as seen in the Example 2.3.1 $|\mathcal{T}_2^2| = \binom{7}{1} = 7$.*

To see that the bounds presented in Theorem 2.2.6 in fact present the exact value for perfect codes we see first the following lemma.

Lemma. 2.3.2. *Let C be a linear perfect binary code with packing radius $\delta_1 = \left\lfloor \frac{\mathrm{d}_1(C)-1}{2} \right\rfloor$. Then*

$$S_{\delta_1+1}(\mathbf{0}) \cap B_{\delta_1+1}(\mathbf{c}_1) \cap B_{\delta_1+1}(\mathbf{c}_2) = \varnothing,$$

for any different $\mathbf{c}_1, \mathbf{c}_2 \in C$.

Proof. First we notice that if $\mathrm{d} := d_1(C)$ is even the result is immediate since in this case $\delta_1 < \dfrac{d-1}{2}$. Here, if we assume the statement is not true, then there exists $v \in S_{\delta_1+1}(\mathbf{0}) \cap B_{\delta_1+1}(\mathbf{c}_1) \cap B_{\delta_1+1}(\mathbf{c}_2)$. Then since δ_1 is the packing radius of C we can suppose without loss of generality(*wlog*) that $\mathbf{c}_1$ is the only codeword that is at distance δ_1 of $\mathbf{v}$, then we have $\mathrm{d}(\mathbf{v}, \mathbf{c}_1) = \delta_1$, $\mathrm{d}(\mathbf{v}, \mathbf{c}_2) = \delta_1 + 1$, therefore by means of the triangular inequality, which is true also since d is a distance, we get that:

$$\mathrm{d}(\mathbf{c}_1, \mathbf{c}_2) \leqslant \mathrm{d}(\mathbf{c}_1, \mathbf{v}) + \mathrm{d}(v, \mathbf{c}_2) = \delta_1 + \delta_1 + 1 = 2\delta_1 + 1 < 2\left(\frac{\mathrm{d}-1}{2}\right) + 1 = \mathrm{d}$$

Then, $\mathrm{d}(\mathbf{c}_1, \mathbf{c}_2) < \mathrm{d}$. This contradicts d being the minimum distance of C.

In general, there is the chance of course that d is odd. In this case, we need to check what happens in each scenario. We are going to show that if there is a vector $\mathbf{v}$ in the given intersection, then there exists also a $\mathbf{v}'$ in the same intersection and depending on the distance between both we prove there exists, as well, a vector $\mathbf{v}^*$ which is at a forbidden distance considering that δ_1 is the packing radius of C a perfect code. Then suppose again the statement is not true and there exists $\mathbf{v} \in S_{\delta_1+1}(\mathbf{0}) \cap B_{\delta_1+1}(\mathbf{c}_1) \cap B_{\delta_1+1}(\mathbf{c}_2)$, and again wlog $\mathbf{v} \in B_{\delta_1}(\mathbf{c}_1)$, this means there exists too $\mathbf{v}' \in S_{\delta_1+1}(\mathbf{0}) \cap B_{\delta_1+1}(\mathbf{c}_1) \cap B_{\delta_1+1}(\mathbf{c}_2)$ different from $\mathbf{v}$, but with $\mathbf{v}' \in B_{\delta_1}(\mathbf{c}_2)$. In this case, we have $\mathrm{d}(\mathbf{v}, \mathbf{c}_1) = \delta_1 = \mathrm{d}(\mathbf{v}', \mathbf{c}_2)$ and $\mathrm{d}(\mathbf{v}, \mathbf{c}_2) = \mathrm{d}(\mathbf{v}', \mathbf{c}_1) = \delta_1 + 1$. In particular, both $\mathbf{v}, \mathbf{v}' \in S_{\delta_1+1}(\mathbf{0})$, this means both have the same weight and thus for them to be different, they need to differ in an even number of coordinates, said in other words $\mathrm{d}(\mathbf{v}, \mathbf{v}') \equiv 0 \mod 2$. Of course since they are different we have $\mathrm{d}(\mathbf{v}, \mathbf{v}') \neq 0$, then we will see what happens when the distance between them is 2 and when it is 4, that allows us to tackle any possibility from there on. We denote by $\mathrm{d}\,|_X\,(\cdot,\cdot)$, the distance between the codewords calculated only considering the coordinates on X.

- **d(v,v')=2**. If it is so, then there are only two coordinates in which both vectors differ, say p_1, p_2, with $p_1 \in \text{Supp}(\mathbf{v}) \setminus \text{Supp}(\mathbf{v}')$ and $p_2 \in \text{Supp}(\mathbf{v}') \setminus \text{Supp}(\mathbf{v})$. Then we can write $v_{p_1} v_{p_2} = 10$, $v'_{p_1} v'_{p_2} = 01$. Now we analyze what happens with the distances from both vectors with respect to one of the codewords, let's say $\mathbf{c}_1$. This distance can be decomposed as the distance calculated in the coordinates corresponding to the intersection of the support of both vectors, plus the distance in the other coordinates. Then let's first notice that in this scenario $\text{d}|_{\text{Supp}(\mathbf{v})\cap\text{Supp}(\mathbf{v}')}(\mathbf{v}, \mathbf{c}_1) = \text{d}|_{\text{Supp}(\mathbf{v})\cap\text{Supp}(\mathbf{v}')}(\mathbf{v}', \mathbf{c}_1)$, $\text{d}(\mathbf{v}, \mathbf{c}_1) = \delta_1$, and $\text{d}(\mathbf{v}', \mathbf{c}_1) = \delta_1 + 1$. Being this, we have four options for those coordinates in c_1 as follows:

  ***1)* 00:** $\text{d}|_{\text{Supp}(\mathbf{v})\cap\text{Supp}(\mathbf{v}')}(\mathbf{v}, \mathbf{c}_1) = \delta_1 - 1$ and then $\text{d}(\mathbf{v}, \mathbf{c}_1) = \delta_1$, but $\text{d}(\mathbf{v}', \mathbf{c}_1) = \delta_1$, which contradicts that δ_1 is 1-st the packing radius of C (from now on written *CPR*).
  ***2)* 01:** $\text{d}|_{\text{Supp}(v)\cap\text{Supp}(\mathbf{v}')}(\mathbf{v}, \mathbf{c}_1) = \delta_1 - 2$, then $\text{d}(\mathbf{v}, \mathbf{c}_1) = \delta_1$, but $\text{d}(\mathbf{v}', \mathbf{c}_1) = \delta_1 - 2$, this CPR.
  ***3)* 10:** $\text{d}|_{\text{Supp}(\mathbf{v})\cap\text{Supp}(\mathbf{v}')}(\mathbf{v}, \mathbf{c}_1) = \delta_1$, then $\text{d}(\mathbf{v}, \mathbf{c}_1) = \delta_1$, but $\text{d}(\mathbf{v}', \mathbf{c}_1) = \delta_1 + 2 \neq \delta_1 + 1$,, which contradicts the given hypothesis.
  ***4)* 11:** $\text{d}|_{\text{Supp}(v)\cap\text{Supp}(\mathbf{v}')}(\mathbf{v}, \mathbf{c}_1) = \delta_1 - 1$, then $\text{d}(\mathbf{v}, \mathbf{c}_1) = \delta_1 = \text{d}(\mathbf{v}', \mathbf{c}_1)$ this CPR.

- **d(v,v')=4** In this scenario there are four coordinates in which both vectors differ, say p_1, p_2, p_3 and p_4. To construct $\mathbf{v}^*$ we consider half of the coordinates in $\text{Supp}(\mathbf{v}) \setminus \text{Supp}(\mathbf{v}') \cup \text{Supp}(\mathbf{v}') \setminus \text{Supp}(\mathbf{v})$ trying to get half from each of these sets $\text{Supp}(\mathbf{v}) \setminus \text{Supp}(\mathbf{v}')$ and $\text{Supp}(\mathbf{v}') \setminus \text{Supp}(\mathbf{v})$ to balance the distance of $\mathbf{v}^*$ from each $\mathbf{v}$ and $\mathbf{v}$', whenever possible.

  By way of illustration, let's suppose $p_1, p_2 \in \text{Supp}(\mathbf{v}) \setminus \text{Supp}(\mathbf{v}')$ and $p_3, p_4 \in \text{Supp}(\mathbf{v}') \setminus \text{Supp}(\mathbf{v})$. Then we consider for $\mathbf{v}^*$ out of them only $p_1, p_3 \in \text{Supp}(\mathbf{v}^*)$, that is, $\mathbf{v}^*$ is 1010, $v_{p_1} v_{p_2} v_{p_3} v_{p_4} = 1100$, and $v'_{p_1} v'_{p_2} v'_{p_3} v'_{p_4} = 0011$ in those four coordinates. Then with respect to the codeword $\mathbf{c}_1$ we have the options presented in the next table.

  By hypothesis, we suppose $\mathbf{v}$ is a vector in the intersection at distance δ_1 from $\mathbf{c}_1$, then what changes when considering the different options p_1 through p_4 for $\mathbf{c}_1$ is $\text{d}(\mathbf{v}, \mathbf{c}_1)\ |_{\text{Supp}(\mathbf{v})\cap\text{Supp}(\mathbf{v}')}$ and therefore $\text{d}(\mathbf{v}', \mathbf{c}_1)$, in general, we just check the coordinates for each case. In Table 5 we can see that for any given possibility of $\mathbf{c}_1$

$c_{1_{p_1}}$	$c_{1_{p_2}}$	$c_{1_{p_3}}$	$c_{1_{p_4}}$	$\mathrm{d}(\mathbf{v},\mathbf{c}_1)\mid_{\mathrm{Supp}(\mathbf{v})\cap\mathrm{Supp}(\mathbf{v}')}$	$\mathrm{d}(\mathbf{v},\mathbf{c}_1)$	$\mathrm{d}(\mathbf{v}',\mathbf{c}_1)$
0	0	0	0	δ_1-2	δ_1	δ_1
0	0	0	1	δ_1-3	δ_1	δ_1-2
0	0	1	0	δ_1-3	δ_1	δ_1-2
0	0	1	1	δ_1-4	δ_1	δ_1-4
0	1	0	0	δ_1-1	δ_1	$\delta_1+2\neq\delta_1+1$
0	1	0	1	δ_1-2	δ_1	δ_1
0	1	1	0	δ_1-2	δ_1	δ_1-2
0	1	1	1	δ_1-3	δ_1	δ_1
1	0	0	0	δ_1-1	δ_1	$\delta_1+2\neq\delta_1+1$
1	0	0	1	δ_1-2	δ_1	δ_1
1	0	1	0	δ_1-2	δ_1	δ_1
1	0	1	1	δ_1-3	δ_1	δ_1-2
1	1	0	0	δ_1	δ_1	$\delta_1+4\neq\delta_1+1$
1	1	0	1	δ_1-1	δ_1	$\delta_1+2\neq\delta_1+1$
1	1	1	0	δ_1-1	δ_1	$\delta_1+2\neq\delta_1+1$
1	1	1	1	δ_1-2	δ_1	δ_1

Table 5 – Scenarios for the example with $\mathrm{d}(\mathbf{v},\mathbf{v}')\equiv 0 \mod 2$

there is not a chance such that $\mathrm{d}(\mathbf{v},\mathbf{c}_1)=\delta_1$ and $\mathrm{d}(\mathbf{v}',\mathbf{c}_1)=\delta_1+1$, having at the same time $\mathrm{d}(\mathbf{v},\mathbf{v}')\equiv 0 \mod 2$; this would mean that the analyzed intersection is not empty. In particular, we can mention that the last column in the table allows us to discard each case as follows:

- Having $\mathrm{d}(\mathbf{v}',\mathbf{c}_1)\leqslant\delta_1$ contradicts δ_1 being the packing radius of the code (CPR).
- Any distance further from δ_1+1 is not possible, as we are considering balls with such a radius.

This means that the only feasible option would have been a distance of exactly δ_1+1. The same could be checked with respect to $\mathbf{c}_2$ and we can say that under the given conditions, for the example taken into consideration, the lemma holds.

Now we check the general case. If we want $\mathbf{v}^*$ to be as much as possible in the middle between $\mathbf{v}$ and $\mathbf{v}$', and different from both, then among the four coordinates studied we can suppose without loss of generality that $p_1,p_2\in\mathrm{Supp}(\mathbf{v})\setminus\mathrm{Supp}(\mathbf{v}')$ and $p_3,p_4\in\mathrm{Supp}(\mathbf{v}')\setminus\mathrm{Supp}(\mathbf{v})$, just by rearranging the coordinates suitably. This leaves us with four possible $\mathbf{v}^*$s, that is, 1010, 1001, 0110, 0101.

Then for each of these options, we check the distances with respect to all the possibilities for $\mathbf{c}_1$. All the possible matches that would lead to a candidate being outside

$c_{1_{p_1}}$	$c_{1_{p_2}}$	$c_{1_{p_3}}$	$c_{1_{p_4}}$	$\mathbf{v}^*$ =1010	$\mathbf{v}^*$ =1001	$\mathbf{v}^*$ =0110	$\mathbf{v}^*$ =0101
0	0	0	0	δ_1	δ_1	δ_1	δ_1
0	0	0	1	δ_1	$\delta_1 - 2$	δ_1	$\delta_1 - 2$
0	0	1	0	$\delta_1 - 2$	δ_1	$\delta_1 - 2$	$\delta_1 - 2$
0	0	1	1	$\delta_1 - 2$	$\delta_1 - 2$	$\delta_1 - 2$	$\delta_1 - 2$
0	1	0	0	$\boldsymbol{\delta_1 + 2}$	$\boldsymbol{\delta_1 + 2}$	δ_1	δ_1
0	1	0	1	$\boldsymbol{\delta_1 + 2}$	δ_1	δ_1	$\delta_1 - 2$
0	1	1	0	δ_1	$\boldsymbol{\delta_1 + 2}$	$\delta_1 - 2$	δ_1
0	1	1	1	δ_1	δ_1	$\delta_1 - 2$	$\delta_1 - 1$
1	0	0	0	$\boldsymbol{\delta_1 + 1}$	δ_1	$\boldsymbol{\delta_1 + 2}$	$\boldsymbol{\delta_1 + 2}$
1	0	0	1	$\boldsymbol{\delta_1 + 1}$	$\delta_1 - 2$	$\boldsymbol{\delta_1 + 2}$	δ_1
1	0	1	0	$\delta_1 - 2$	δ_1	δ_1	$\boldsymbol{\delta_1 + 2}$
1	0	1	1	$\delta_1 - 2$	$\delta_1 - 2$	δ_1	δ_1
1	1	0	0	$\boldsymbol{\delta_1 + 2}$	$\boldsymbol{\delta_1 + 2}$	$\boldsymbol{\delta_1 + 2}$	$\boldsymbol{\delta_1 + 2}$
1	1	0	1	$\boldsymbol{\delta_1 + 2}$	δ_1	$\boldsymbol{\delta_1 + 2}$	δ_1
1	1	1	0	δ_1	$\boldsymbol{\delta_1 + 2}$	δ_1	$\boldsymbol{\delta_1 + 2}$
1	1	1	1	δ_1	δ_1	δ_1	δ_1

Table 6 – $\mathrm{d}(\mathbf{v}^*, \mathbf{c}_1)$ considering $[(\mathrm{Supp}(\mathbf{v}') \setminus \mathrm{Supp}(\mathbf{v})) \cup (\mathrm{Supp}(\mathbf{v}) \setminus \mathrm{Supp}(\mathbf{v}'))]$

the intersection, mentioned in the lemma, are highlighted (also those with distance $\delta_1 + 1$ since they wouldn't let us get to a contradiction and there are other options on the same line that would indeed); but it is to notice that for every choice of $\mathbf{c}_1$ there is a choice for $\mathbf{v}^*$ to which $\mathrm{d}(\mathbf{v}^*, \mathbf{c}_1) \leqslant \delta_1$, meaning there is always a possible $\mathbf{v}^*$ we can choose from so that it contradicts the fact of being δ_1 the packing radius, since the same $\mathbf{v}^*$ will have a similar behavior with respect to $\mathbf{c}_2$, by construction.

The only codeword for which no such a contradiction would be found is when $\mathbf{c}_1 \mid_{\{p_1,p_2,p_3,p_4\}} = 1100$, but then again this choice is not possible because under such circumstances the coordinates that count for $\mathrm{d}(\mathbf{v}, \mathbf{c}_1)$ would be in $\mathrm{Supp}(\mathbf{v}) \cap \mathrm{Supp}(\mathbf{v}')$ and in consequence it would not be possible to increase the distance of $\mathbf{c}_1$ with respect to $\mathbf{v}$' and at the same time changing the chosen four coordinates, that is, there would be no way to obtain $\mathrm{d}(\mathbf{v}', \mathbf{c}_1) = \delta_1 + 1$, which contradicts the general hypothesis. Therefore, in general, given the circumstances of the lemma there does not exist a vector in the intersection and in consequence it is empty. Of course, we could check the same for the cases $\mathrm{d}(\mathbf{v}, \mathbf{v}') = 6, 8,$ *etc.* But in those scenarios, it is even clearer that we could find a vector that would contradict the condition of being δ_1 1-st the packing radius of the code. T herefore, there should not be an intersection between balls with that radius.

□

Remark 2.3.2. *It could be thought that a generalization of the proposition of the previous lemma would be possible, i.e., for* $k \leqslant \dim(C)$ *and* $\{\mathbf{c}_1, \ldots, \mathbf{c}_{k+1}\} \subset C \setminus \{\mathbf{0}\}$

$$S_{\delta_k+1}(\mathbf{0}) \cap B_{\delta_k+1}(\mathbf{c}_1) \cap \cdots \cap B_{\delta_k+1}(\mathbf{c}_{k+1}) = \varnothing$$

However, this property where the intersections of balls centered at non-zero codewords with the sphere centered at the null-vector are mutually disjoint cannot be established in general. For instance, let us consider C *the Hamming* $[7, 4, 3]$*-code and let's consider* $k = 1$, $\mathbf{c}_1 = (0011010)$, $\mathbf{c}_2 = (1101000)$, $\mathbf{c}_3 = (0100011)$*, then we see that for this code* $\delta_2 := \lfloor \frac{d_2(C) - 1}{2} \rfloor = 2$*. And yet*

$$S_{\delta_2+1}(\mathbf{0}) \cap B_{\delta_2+1}(\mathbf{c}_1) \cap B_{\delta_2+1}(\mathbf{c}_2) \cap B_{\delta_2+1}(\mathbf{c}_3) = \{(0101010)\} \neq \varnothing.$$

Also we could try to consider the packing radius for codes that are not perfect and see if the property could be extended to their covering radius, but again this is not true in general. Next, there is a counterexample. Let us consider C *the code generated by the matrix*

$$G = \begin{pmatrix} 10010110 \\ 01010101 \\ 00110011 \\ 00001111 \end{pmatrix},$$

this is an $[8, 4, 4]$ *binary code with covering radius 2 and packing radius 1, then* $\delta_1 + 1 = 2$*. For this code if we take* $\mathbf{c}_1 = (11000011)$, $\mathbf{c}_2 = (10101010)$ *we get*

$$S_{\delta_1+1}(\mathbf{0}) \cap B_{\delta_1+1}(\mathbf{c}_1) \cap B_{\delta_1+1}(\mathbf{c}_2) = \{(10000010)\} \neq \varnothing.$$

All this shows us that the condition of the code to be perfect is mandatory and that taking the packing radius makes the expression tight, this means, it cannot be obtained for greater, and by definition not even smaller, values for the radius.

After seeing this condition holds for perfect codes, we can then state the following theorem that provides us with a closed expression for $\mathrm{PCD}_{\leqslant e}(C)$.

Proposition. 2.3.3. *Let* C *be a linear binary perfect code with packing radius* δ_1*, then if* $e = \delta_1 + 1$ *the first step of the bound is in fact the exact value for* $\mathrm{PCD}_{\leqslant e}(C)$*. Here:*

$$\mathrm{PCD}_{\leqslant e}(C) = \frac{\sum_{t=0}^{\delta_1} |S_t(\mathbf{0})| + \left(|S_e(\mathbf{0})| - |C_{d_1(C)}||S_e(\mathbf{0}) \cap B_e(\mathbf{c}_{d_1(C)})|\right)}{|B_e(\boldsymbol{c})|}$$

$$= \frac{|B_{\delta_1}(\mathbf{0})| + \left(|S_e(\mathbf{0})| - |C_{d_1(C)}||S_e(\mathbf{0}) \cap B_e(\mathbf{c}_{d_1(C)})|\right)}{|B_e(\boldsymbol{c})|}$$

Proof. Let us consider $C_{\mathrm{d}_1(C)}$ the set of words of minimum weight, then by Lemma 2.3.2, for every $\mathbf{c}_1, \mathbf{c}_2 \in C_{\mathrm{d}_1(C)}$, with $\mathbf{c}_1 \neq \mathbf{c}_2$, $B_{\delta_1}(\mathbf{c}_1) \cap B_{\delta_1}(\mathbf{c}_2) = \varnothing$. Since it is true for every pair of codewords, it is also true for the codewords in $C_{\mathrm{d}_1(C)}$. Now here also

$$S_{\delta_1+1}(\mathbf{0}) \cap B_{\delta_1+1}(\mathbf{c}_1) \cap B_{\delta_1+1}(\mathbf{c}_2) = \varnothing,$$

for every $\mathbf{c}_1, \mathbf{c}_2 \in C_{\mathrm{d}_1(C)}$, with $\mathbf{c}_1 \neq \mathbf{c}_2$. This tells us that the other steps in Proposition 2.2.6 are not necessary and the first step instead of a bound is the exact value of $\mathrm{PCD}_{\leqslant e}(C)$. □

Example. For C the binary Golay Code [23,12,7] we have $\delta_1 = 3$. Then for $e = 4$ we get

$$PCD_{\leqslant e}(C) = \frac{|B_3(\mathbf{0})| + (|S_4| - |C_7||S_4(\mathbf{0}) \cap B_4(\mathbf{c}_7)|)}{|B_4(\mathbf{c})|}$$

Then,

$$PCD_{\leqslant e}(C) = \frac{2048 + (8855 - 253 \cdot 35)}{10903} = 0,187838$$

The generalized Hamming weights had already been useful to analyze the error probability of a code over an erasure channel as seen in (SHEN; FU, 2019), but this approach does not work over a binary symmetric channel, because we do not know where the error occurs. Some bounds were established for the i-th Hamming weight itself in (HELLESETH; KLØVE; YTREHUS, 1992), but not much had been said about the error correction capacity a code C has, related to its weight distribution. And it is what has been presented in this chapter.

We have seen how these two figures of merit of C, $\mathrm{PCD}_{\leqslant e}(C)$ and $\mathrm{PED}_{=e}(C)$, introduced in this work, are a meaningful tool to effectively compare codes with equal or similar minimum distance, just as seen in Table 2. In general, due to the inclusion-exclusion principle, these values can be calculated either tightly or exactly, depending on the case.

For perfect codes, we saw in Proposition 2.3.3 that they provide an exact expression for each one of the values, being the first step the only one required. Nonetheless, taking into consideration the generalized Hamming weights, drew our attention to the complete spectrum of the code and how it could affect its ability to correct errors, but on

the way more than that it showed also its potential as a new tool to classify codes, and it is that what the next chapter is about.

2.4 Open question about $\mathrm{PCD}_{\leqslant e}(C)$ and $\mathrm{PED}_{=e}(C)$

In this chapter we obtained bounds for the figures of merit $\mathrm{PCD}_{\leqslant e}(C)$ and $\mathrm{PED}_{=e}(C)$ in the binary case. We would like to investigate later if we can generalize these concepts over $\mathbb{F}_q$, including the result for the intersection of balls presented previously in Proposition 2.2.1. Also to determine if tighter bounds can be established and knowing the number of steps necessary to get to the exact value of $\mathrm{PCD}_{\leqslant e}(C)$ by means of the inclusion-exclusion principle.

3 Generalized Hamming weights as complete invariants of codes

Usually when considering the correction qualities of a code, it is only checked how convenient its classical minimum distance is. However, what we have been seeing along this work is that the capacity a code has to correct a given number of errors does not depend only on this parameter, but more so on its generalized Hamming weights and even its complete weight spectrum. Ever since Wei introduced the concept in (WEI, 1991), the generalized Hamming weights have remained an interesting definition, but its role is not actually understood. Nonetheless, it has been used in other types of codes, rather than linear block codes for which were defined. Rosenthal extended the definition for convolutional codes in (ROSENTHAL; YORK, 1997), Ravagnani in (RAVAGNANI, 2016) for Gabidulin codes in relation to Network coding. Here we present some results that support the idea that the generalized Hamming weights and, even further, the spectrum of the code (which gives the complete weight distribution of the codewords in a code C) is a complete set of invariants, that is, it can determine a code up to equivalence.

It is clear by many examples, that the fact that two codes have the same minimum distance, does not imply any relation, in the matter of equivalence, between such codes. For instance, consider the full ambient space $\mathbb{F}_q^n$ and the code spanned by a weight one vector, both have minimum distance 1, but are clearly non-equivalent in general. Now, what can be said about codes with the same weight enumerator? Is there any between them under this consideration? It can be seen in the classification files found on GitHub or in the following example from (CHEON, 2006) that equal weight polynomial is not enough to say that the codes are equivalent.

Example. 3.0.1. *Given*

$$G_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}, \; G_2 = (I_3 \mid I_3).$$

And say C_i is the code whose generator matrix is G_i. Then, we have that $w_{C_1}(x) = (1+x^2)^3 = 1 + 3x^2 + 3x^4 + x^6 = w_{C_2}(x)$. But, while for every $\mathbf{x}_1, \mathbf{x}_2 \in C_2, \mathbf{x}_1 \neq \mathbf{x}_2$, of

weight 2*, we get* $\mathrm{wt}(\mathbf{x}_1+\mathbf{x}_2)=4$*; on* C_1 *there exist* $\mathbf{x}_1', \mathbf{x}_2'$ *of weight* 2 *and* $\mathrm{wt}(\mathbf{x}_1'+\mathbf{x}_2')=2$*. For instance, consider* $\mathbf{x}_1' = G_1[1,:],\ \mathbf{x}_2' = G_1[2,:]$.

This example tells us that having codes with equal weight polynomials is not enough to say the codes are isomorphic. But, is there a way we could affirm something like that, considering the support of all the subcodes of a given binary code C? This is the question that motivates us in this chapter. Even though we do not have a final answer in the general case, there is strong evidence that suggests that what we call the spectrum of the code is a complete set of invariants, meaning it can be used to classify binary codes. So let us begin first introducing this concept of the spectrum of a code.

3.1 Introduction

To get information about the capacity a code has to correct errors, and what really defines it, we have noticed that not only the generalized Hamming weights are important, but the weight spectrum itself, i.e., the matrix of the distribution of supports for the subcodes of a given code C. We recall the spectrum definition as given in Definition 2.2.4 where $Spec(C) = (A_i^j)$, and

$$A_i^j := |\{D \subset C : \dim(D) = i \text{ and } ||D|| := |\mathrm{Supp}(D)| = j\}|.$$

We shall represent $Spec(C)$ by a matrix, as in this example.

Example. 3.1.1. *If* C *has a generator matrix*

$$G := \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \text{ then } Spec(C) := \begin{pmatrix} 1 & 2 & 2 & 1 & 1 \\ 0 & 0 & 2 & 1 & 4 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

In fact, we have $C := \{00000,\ 11010,\ 00111,\ 00010,\ 11101,\ 00101,\ 11000,\ 11111\}$. *Counting the number of nonzero codewords of a given weight we get the first row of* $Spec(C)$*. The number of 2-dimensional subspaces of* C *is given by the q-binomial coefficient when* $q = 2$, $\binom{3}{2}_2 = 7$*, which are:*

- $\langle 11010, 00010\rangle$ *with support size* 3
- $\langle 00111, 00010\rangle$ *with support size* 3

- $\langle 11101, 00101\rangle$ *with support size* 4
- $\langle 11010, 00111\rangle$ *with support size* 5
- $\langle 11010, 00101\rangle$ *with support size* 5
- $\langle 00111, 11000\rangle$ *with support size* 5
- $\langle 00010, 11101\rangle$ *with support size* 5

From this we obtain the second row of $Spec(C)$ and finally its last one comes from the fact that $|\mathrm{Supp}(C)| = |\{1, 2, 3, 4, 5\}|$.

From here on, in this chapter, our discussion is focused on linear binary codes, unless otherwise stated. We next present a couple of examples that allows us to think that given a pair of codes with equal spectrum, if we can find a couple of codewords with the same weight in a respective basis for these codes, and they are removed, the effect over the spectrum of such codes is the same, whether the codewords are equal or not, as long as they have the same weight. And this gives us some ideas on how to associate what we call spectrum preserving bases with the notion of equivalence. Then let us consider

$$G_1 := \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad G_2 := \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

and define them to be generator matrices for $C_i, i = 1, 2$, respectively. They are equivalent codes with $\sigma' = (175286)(34) \in \mathrm{Sym}(8)$. Their spectrum matrix is

$$Spec(C_i) = \begin{pmatrix} 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, i = 1, 2.$$

We can define $\mathbf{c}_1 := (00001111)$, $\mathbf{c}_2 := (01010101)$, with support $\mathrm{Supp}(\mathbf{c}_1) = \{5, 6, 7, 8\}$, $\mathrm{Supp}(\mathbf{c}_2) = \{2, 4, 6, 8\}$. If we include $\mathbf{c}_i$ into the considered basis of C_i, we get C generated by the matrix,

$$G := \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

We notice that $\sigma' = (175286)(34) = (17)(15)(12)(18)(16)(34)$ and removing those transpositions with coordinates on $\mathrm{Supp}(\mathbf{c}_1) \cup \mathrm{Supp}(\mathbf{c}_2)$ we get $\sigma = (1) = Id$ as expected.

This example puts interest into looking how to work with the bases of the codes to determine some relationship between the codes as we are looking for. That is what motivates us to give the next definition that shows a condition we would expect to be fulfilled by codes with equal spectrum.

3.2 Spectrum preserving bases

Definition. 3.2.1 (Codes with spectrum preserving bases)**.** *Given C, C' k-dimensional binary codes of length n. We say that C, C' admit **spectrum preserving bases** if for every $\mathcal{B} = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ a basis of C, there exists $\mathcal{B}' = \{\mathbf{c}'_1, \ldots, \mathbf{c}'_k\}$ a basis of C', such that $\mathrm{wt}(\mathbf{c}_i) = \mathrm{wt}(\mathbf{c}'_i)$ and $Spec\langle \mathbf{c}_1, \ldots, \mathbf{c}_i \rangle = Spec\langle \mathbf{c}'_1, \ldots, \mathbf{c}'_i \rangle$, for every $i = 1, \ldots, k$.*

The next proposition tells us that the subspaces of codes with equal spectrum are linked pairwise, so to say. This means that for every subspace in one of the codes, there must be a subspace in the other, both, with equal spectrum.

Proposition 3.2.1. *Let $C,\ C' \leqslant \mathbb{F}_2^n$ be codes with spectrum preserving bases. Then, for every linear subcode $S \leqslant C$, there exists $S' \leqslant C'$, such that $Spec(S) = Spec(S')$.*

Proof. Given C, C' binary k-dimensional codes of length n that admit spectrum preserving bases. For any $S \leqslant C$, given a basis of S it can be extended to a basis $\mathcal{B}$ of C. Therefore, wlog we can assume the first $\dim(S)$ vectors in $\mathcal{B}$ to constitute a basis of S and, by definition, there exists a corresponding spectrum preserving basis $\mathcal{B}'$ of C'. For this $\mathcal{B}'$ we can also take the first $\dim(S)$ vectors of $\mathcal{B}'$ to span a subspace of C' that, by definition again, has the same spectrum as S and this subspace is the S' we needed. □

The next lemma lets us know that whenever a pair of codes have equal spectrum, they must admit what we call spectrum preserving bases.

Theorem 3.2.2. *Given C, C' k-dimensional binary codes of length n. They admit spectrum preserving bases if, and only if, $Spec(C) = Spec(C')$.*

Proof. The first implication is an immediate consequence of Definition 3.2.1, since we have that it is true for every $i := 1, \ldots, k$, it is in particular true for $i = k$. For the converse, since we suppose both C and C' have the same spectrum, we can fix a basis for C, let us say $\mathcal{B} := \{\mathbf{c}_1, \ldots, \mathbf{c}_{k-1}, \mathbf{c}_k\}$ and we can choose $\mathbf{c}'_1 \in C'$, with $\mathrm{wt}(\mathbf{c}_1) = \mathrm{wt}(\mathbf{c}'_1)$ and it is clear that having the same weight, the spectrum of the subcodes spanned, by these vectors respectively, will be the same. Now suppose the statement is true up to dimension $l \leqslant k - 1 < k$, i.e., if $Spec(\langle \mathbf{c}_1, \ldots, \mathbf{c}_l \rangle) = Spec(\langle \mathbf{c}'_1, \ldots, \mathbf{c}'_l \rangle)$, then $\langle \mathbf{c}_1, \ldots, \mathbf{c}_l \rangle$ and $\langle \mathbf{c}'_1, \ldots, \mathbf{c}'_l \rangle$ admit spectrum preserving basis. Let us prove for k, consider the hypothesis to be true for $l = k - 1$, then name the $(k-1)$-dimensional subspaces $S := \langle \mathbf{c}_1, \ldots, \mathbf{c}_{k-1} \rangle \leqslant C$ and $S' := \langle \mathbf{c}'_1, \ldots, \mathbf{c}'_{k-1} \rangle \leqslant C'$ for which $Spec(S) = Spec(S')$ and that admit spectrum preserving basis. We have by hypothesis that C and C' have the same spectrum, then the weight distribution of the words in C and $C \setminus S$, and C' and $C' \setminus S'$ are equal, respectively. Moreover, $\mathbf{c}_k \in C \setminus S$ and there exists $\mathbf{c}'_k \in C' \setminus S'$ with $\mathrm{wt}(\mathbf{c}_k) = \mathrm{wt}(\mathbf{c}'_k)$ that completes a basis for C'(any l.i with $\{c'_1, \ldots, c'_{k-1}\}$ of weight $\mathrm{wt}(c_k)$) and clearly also $Spec(C) = Spec(\langle \mathbf{c}_1, \ldots, \mathbf{c}_{k-1}, \mathbf{c}_k \rangle) = Spec(\langle \mathbf{c}'_1, \ldots, \mathbf{c}'_{k-1}, \mathbf{c}'_k \rangle) = Spec(C')$, ending this way our proof. □

Remark 3.2.3. *In this proof the existence of each c'_i with the same weight as c_i is a consequence of the hypothesis that $Spec(C) = Spec(C')$. On the other hand, that each one fulfills the condition that the spanned vector spaces have the same spectrum comes additionally from noticing that Definition 2.2.4 implies an appropriate organization of the support of the subspaces. This in order to guarantee the equality of the higher dimension subcodes. Thus, even though having subspaces with equal spectrum is not sufficient to guarantee that the codes have equal spectrum too, it is indeed necessary. In this sense, Definition 2.2.4 lets us assert that if for a given dimension $l \leqslant k = \dim(C) = \dim(C')$ there do not exist l-dimensional subspaces $S \leqslant C$ and $S' \leqslant C'$ with $Spec(S) = Spec(S')$, then $Spec(C) \neq Spec(C')$.*

As an illustration on how to obtain these spectrum preserving bases, we show a couple of examples.

I.) Let us consider the codes with generator matrices

$$G_1 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \; G_2 := \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

which are equivalent and have a spectrum matrix

$$\begin{pmatrix} 1 & 3 & 3 & 0 \\ 0 & 0 & 4 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

First let's obtain the three spectrum matrices of the subcodes generated by the vectors of the considered basis of C_1, the code generated by G_1, that we need to determine B'. These are

$$Spec(\langle\{(1000)\}\rangle) = (1\ 0\ 0\ 0\ 0\ 0),$$

$$Spec(\langle\{(1000),(0101)\}\rangle) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

,

$$Spec(\langle\{(1000),(0101),(0011)\}\rangle) = \begin{pmatrix} 1 & 3 & 3 & 0 \\ 0 & 0 & 4 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Now if we notice from the vectors in the generator matrix G_1, we see there are two vectors of weight 2 and one of weight 1. Then, we need to work with the vectors of weight 1 and 2 of the code generated by G_2. These sets are $\{0001\}$, $\{1010, 0110, 1100\}$, respectively. Then we start constructing the corresponding basis B' with respect to the basis with vectors in G_1, by taking the first, and in this case only, vector of weight 1 in C_2, the code generated by G_2, i.e., (0001). Of course then for dimension 1 we get as

$$Spec(\langle\{(0001)\}\rangle) = (100000).$$

Next we need to choose one vector of weight two that, when joined to the basis, spans a code with the same spectrum as that shown for dimension two for the code C_1. Then here there are three possible combinations,

$1 : \{(0001), (1010)\}, 2 : \{(0001), (0110)\}, 3 : \{(0001), (1100)\}$, which we call the options 1, 2 and 3. The respective spectrum matrices are,

$$1 : \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, 2 : \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, 3 : \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

As we can notice they are all the same, which means there is not a unique option we can choose. And therefore, any linearly independent vector in the set of 2-weighted vectors will complete the basis and span the full code C_2, which in consequence has the same spectrum as C_1. Then, for instance, an appropriate B' obtained using this procedure is $\{(0001),\ (1010),\ (0110)\}$.

II.) For the second example, let us consider the codes with generator matrices:

$$G_1 := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix},\ G_2 := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

which are equivalent and have a spectrum matrix

$$\begin{pmatrix} 1 & 1 & 2 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

First let's obtain the three spectrum matrices of C_1, the code generated by G_1, that we need to determine B'. These are

$$Spec(\langle\{(100000)\}\rangle) = (1\ 0\ 0\ 0\ 0\ 0),$$

$$Spec(\langle\{(100000), (010001)\}\rangle) = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix},$$

$$Spec(\langle\{(100000), (010001), (000011)\}\rangle) = \begin{pmatrix} 1 & 1 & 2 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Now if we notice from the vectors in the generator matrix G_1, we see there is one vector of weight 3, one of weight 2 and one of weight 1. Then according to the Definition 3.2.1, we need to work with the vectors of weight 1, 2 and 3 of the code generated by G_2. These sets are $\{(000001)\}$, $\{(100010)\}$, $\{(011100), (100011)\}$,

respectively. Then we start constructing the corresponding basis B' with respect to the base with vectors in G_1, by taking the first, and in this case only, vector of weight 1 in C_2, here the code generated by G_2, i.e., (000001). Of course then for dimension 1 we get as

$$Spec(\langle\{(000001)\}\rangle) = (100000).$$

Next we need to choose one vector of weight three, if we want to preserve the order of the rows in G_1. We choose that vector so that, when joined to the basis, it spans a code with the same spectrum as that shown for dimension two for the code C_1. Then here there are two possible combinations, $1 : \{(000001), (011100)\}$, $2 : \{(000001), (100011)\}$, which we call the options 1 and 2. The respective spectrum matrices are

$$1 : \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \; 2 : \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

We can see the only possible choice is the option 1, that preserves the spectrum, as required. It does exists because codes C_1 and C_2 are equivalent. Finally just add the only weight 2 codeword, i.e., (100010) obtain a basis of the code C_2 and consequently the full spectrum matrix. Then here B' turns to be unique and is $\{(000001),\ (011100),\ (100010)\}$.

3.3 Spectrum and equivalence

The following theorem relates equivalent codes with the property of having equal spectrum.

Theorem 3.3.1. *Equivalent binary codes have the same weight spectrum.*

Proof. Let $C, C' \subseteq \mathbb{F}_2^n$ be two equivalent codes. Then, by definition of equivalence, there exists $\sigma \in Sym(n)$, such that $\sigma(C) = C'$. Now since by Theorem 1.2.1 the support of a code is determined by its generator matrix and since C, C' are equivalent, we have that

$$(\forall S \leqslant C)(\sigma(S) \leqslant C' \text{ and } ||S|| = ||\sigma(S)||).$$

Analogously,

$$(\forall S' \leqslant C')(\sigma^{-1}(S') \leqslant C \text{ and } ||S'|| = ||\sigma^{-1}(S')||).$$

This means there is a bijection between the subspaces of C and C', by means of the isomorphism induced by σ, that also preserves the support of the subspaces that are associated, and since it is true for every subspace we have in consequence that $Spec(C) = Spec(C')$. □

Exploring the relationship between spectrum and equivalence has been the purpose of this chapter and so far we have found the following implications

$$\textbf{Equiv. codes} \underset{\text{Conjecture}}{\overset{\text{Th. 3.3.1}}{\rightleftharpoons}} \textbf{Equal spec.} \overset{\text{Th. 3.2.2}}{\longleftrightarrow} \textbf{Admit spec. pres. basis}$$

The following example illustrates a scenario in which the converse of Theorem 3.3.1 holds. In this case, it is possible to obtain, from an isomorphism between subspaces, an isomorphism between the codes themselves.

$$GG_1 := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \; GG_2 := \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

and define GG_i to be a generator matrix for $CC_i,\ i = 1, 2$. Both codes have the spectrum matrix

$$Spec(CC_i) = \begin{pmatrix} 0 & 1 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, i = 1, 2.$$

These codes are equivalent, but not equal. In fact $CC_1 \equiv CC_2$, with $\sigma' = (17423)$ and if we include into their basis $\mathbf{c}^*$ =(0 1 1 0 0 1 1 1) a vector of weight 5 we get respectively C_1 and C_2 with the next generator matrices

$$G_1 := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \; G_2 := \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

We can check by simple inspection even, that $C_1 \equiv C_2$ with $\sigma = (14) \in Sym(8)$. Both codes have the spectrum

$$Spec(C_i) = \begin{pmatrix} 0 & 1 & 1 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, i = 1, 2.$$

Now we notice that $\text{Supp}(\mathbf{c}^*) = \{2, 3, 6, 7, 8\}$ and $\sigma' = (17423) = (17)(14)(12)(13)$, if we remove the transpositions with coordinates on $\text{Supp}(c^*)$ we get $\sigma = (14)$, which makes

$C1 \equiv C_2$. Unfortunately this is not always possible, then this technique does not allow us to prove in general the converse of Theorem 3.3.1.

In fact, Theorem 3.3.1 is rather simple and can be generalized to q-ary codes. We are rather interested in its reciprocal: does the generalized weight spectrum characterizes a code up to equivalence? This would be a remarkable result, since considering less than the spectrum is not sufficient to characterize a code: as we shall see, neither the weight distribution (or Mac Williams polynomial) nor the generalized weigh hierarchy are sufficient to ensure codes equivalence.

We devote the next section to explore this conjecture.

3.4 Shortcuts for classification by exhaustion

The following lemma tells us a relation between equivalent codes and their duals. And therefore when searching for equivalent codes in a n-dimesional ambient space, it is sufficient to search up to dimension $\frac{n}{2}$ or $\lfloor\frac{n}{2}\rfloor + 1 = \lfloor\frac{n+1}{2}\rfloor$, case n is odd.

Lemma. 3.4.1. *Let $C \subseteq \mathbb{F}_q^n$ and $\sigma \in Sym(n)$. Then $\sigma(C)^\perp = \sigma(C^\perp)$. Or equivalently let $C_1, C_2 \subseteq \mathbb{F}_q^n$ equivalent codes, i.e., there exists $\sigma \in Sym(n)$, such that, $\sigma(C_1) = C_2$. Then $\sigma(C_1)^\perp = C_2^\perp$.*

Proof. Let $\sigma \in Sym(n)$ and $C \subseteq \mathbb{F}_q^n$. Let us prove that $\sigma(C)^\perp = \sigma(C^\perp)$. In fact, for every $\mathbf{x} = (x_1, \ldots, x_n) \in C, \mathbf{y} = (y_1, \ldots, y_n) \in C^\perp$ we have

$$0 = \langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i = \sum_{i=1}^{n} x_{\sigma(i)} y_{\sigma(i)} = \langle \sigma(\mathbf{x}), \sigma(\mathbf{y}) \rangle$$

Then

$$\mathbf{y} \in C^\perp \Leftrightarrow \sigma(\mathbf{y}) \in \sigma(C)^\perp.$$

And in consequence

$$\sigma(C)^\perp = \sigma(C^\perp).$$

□

This lemma then allows us to assert that

$$C \equiv C' \Leftrightarrow C^\perp \equiv C'^\perp.$$

There are two scenarios for which the general assertion, the converse of Theorem 3.3.1, is true. They are the trivial case when dimension is 1, in which the vector spaces, being binary, have only a nonzero codeword and both have the same weight, this leads to being equivalent by having any bijection between the support of both codewords, an induced isomorphism between the vector spaces. And the second case is proven in the following proposition.

Proposition 3.4.1. *Given $C, C' \leqslant \mathbb{F}_2^n$ 2-dimensional spaces with $Spec(C) = Spec(C')$. Then $C \equiv C'$.*

Proof. If C, C' are binary codes of dimension 2, then there exist $\alpha, \beta, \mathbf{x}, \mathbf{x}' \in \mathbb{F}_2^n$ such that $C = \langle \alpha, \mathbf{x} \rangle, C' = \langle \beta, \mathbf{x}' \rangle$ and given the fact that both have the same spectrum, we can suppose without loss of generality that $\mathrm{wt}(\alpha) = \mathrm{wt}(\beta)$ and $\mathrm{wt}(\mathbf{x}) = \mathrm{wt}(\mathbf{x}')$. In general, we would get $C = \{\mathbf{0}, \alpha, \mathbf{x}, \alpha + \mathbf{x}\}, C' = \{\mathbf{0}, \beta, \mathbf{x}', \beta + \mathbf{x}'\}$. In this case, since $Spec(C) = Spec(C')$ necessarily $\mathrm{wt}(\alpha + \mathbf{x}) = \mathrm{wt}(\beta + \mathbf{x}')$. Now, how can this information lead to both codes being equivalent? That is what we pretend to prove next. And we do this by showing there is a partition of the support of the code in terms of the support of the elements of the considered basis. Over $\mathbb{F}_2$ it is easy to check that for every $\mathbf{y}, \mathbf{z} \in \mathbb{F}_2^n, \mathrm{wt}(\mathbf{y} + \mathbf{z}) = |(S(\mathbf{y}) \cup S(\mathbf{z})) \smallsetminus (S(\mathbf{y}) \cap S(\mathbf{z}))|$. Moreover, $\mathrm{wt}(\mathbf{y} + \mathbf{z}) = \mathrm{wt}(\mathbf{y}) + \mathrm{wt}(\mathbf{z}) - 2|S(\mathbf{y}) \cap S(\mathbf{z})|$. This last one equality indicates that under the assumption we have that both α, β and $\mathbf{x}, \mathbf{x}'$ have the same weight, respectively, and having the same spectrum both codes, then it is also true that $\alpha + \mathbf{x}, \beta + \mathbf{x}'$ have the same weight, then $\mathrm{wt}(\alpha + \mathbf{x}) = \mathrm{wt}(\alpha) + \mathrm{wt}(\mathbf{x}) - 2|S(\alpha) \cap S(\mathbf{x})| = \mathrm{wt}(\beta) + \mathrm{wt}(\mathbf{x}') - 2|S(\beta) \cap S(\mathbf{x}')| = \mathrm{wt}(\beta + \mathbf{x}')$. Therefore, $|S(\alpha) \cap S(\mathbf{x})| = |S(\beta) \cap S(\mathbf{x}')|$. But with this we also have that $|S(\alpha) \smallsetminus S(\mathbf{x})| = |S(\beta) \smallsetminus S(\mathbf{x}')|$, because it is already true that $|S(\alpha)| = |S(\beta)|$ and $S(\alpha) = (S(\alpha) \smallsetminus S(\mathbf{x})) \cup (S(\alpha) \cap S(\mathbf{x}))$, then $|S(\alpha) \smallsetminus S(\mathbf{x})| = |S(\beta) \smallsetminus S(\mathbf{x}')|$. The same can be seen with respect to $S(\mathbf{x})$ and $S(\mathbf{x}')$. Then we have that effectively $|S(\alpha) \cup S(\mathbf{x})| = |S(\beta) \cup S(\mathbf{x}')|$ and there is a partition with sizes of the sets corresponding to each other too, i.e.,

$$S(\alpha) \cup S(\mathbf{x}) = (S(\alpha) \smallsetminus S(\mathbf{x})) \cup (S(\alpha) \cap S(\mathbf{x})) \cup (S(\mathbf{x}) \cup S(\alpha)),$$

$$S(\beta) \cup S(\mathbf{x}') = (S(\beta) \smallsetminus S(\mathbf{x}')) \cup (S(\beta) \cap S(\mathbf{x}')) \cup (S(\mathbf{x}') \cup S(\beta)),$$

$$|S(\alpha) \smallsetminus S(\mathbf{x})| = |S(\beta) \smallsetminus S(\mathbf{x}')|,$$

$$|S(\mathbf{x}) \smallsetminus S(\alpha)| = |S(\mathbf{x}') \smallsetminus S(\beta)| \text{ and}$$

$$|S(\alpha) \cap S(\mathbf{x})| = |S(\beta) \cap S(\mathbf{x}')|.$$

Then there is a bijection σ_1 between $S(\alpha) \setminus S(\mathbf{x})$ and $S(\beta) \setminus S(\mathbf{x}')$, a bijection σ_2 between $S(\mathbf{x}) \setminus S(\alpha)$ and $S(\mathbf{x}') \setminus S(\beta)$, and a bijection σ_3 between $S(\mathbf{x}) \cap S(\alpha)$ and $S(\mathbf{x}') \cap S(\beta)$. Any of them in case of being empty turns σ_i to be the identity, and being a partition we can consider then that $\sigma = \sigma_1\sigma_2\sigma_3$ induces an isomorphism between C and C'. □

Theorem 3.4.2. *Two k-dimensional binary codes with same spectrum are equivalent for $k \in \{1, 2, n-2, n-1\}$.*

Proof. The case $k = 1$ is trivial and $k = 2$ was proved in the previous proposition. The cases $k = n-2, n-1$ follow from Lemma 3.4.1. □

3.4.0.1 COMPUTATIONS FOR $\mathbf{n \leqslant 11}$

The next proposition is also useful to reduce the calculations while computing the number of equivalence classes of binary codes of a given length and dimension. By $\mathcal{S}_k$ we denote a set of all the equivalence classes' representatives of k-dimensional subspaces of $\mathbb{F}_2^n$.

Proposition 3.4.3. *Given $\mathcal{S}_k$ be a set of representatives of k-dimensional subspaces of $\mathbb{F}_2^n$. Then for every $C \in \mathcal{S}_k$, any $(k-1)$-dimensional subspace $M \leqslant C$ is equivalent to some $L \in \mathcal{S}_{k-1}$*

Proof. Given $C \in \mathcal{S}_k$. Suppose that for some $M \leqslant C$ with $\dim(M) = k-1$ there is no $L \in \mathcal{S}_{k-1}$ such that $M \equiv L$. This contradicts the fact that $\mathcal{S}_{k-1}$ contains all the equivalence classes' representatives of $(k-1)$-dimensional subspaces of $\mathbb{F}_2^n$. □

This result was important, specially when analyzing $n \geqslant 9$ where the number of possible subspaces grows considerably. This guarantees that the set of representatives of higher dimension may be obtained using those of smaller dimensions. And since we were interested in checking the condition given in the Conjecture 3.4.2 for codes that were not equivalent and had the same spectrum, if they existed, and we could verify that it is not true. This is, whenever the codes had the same spectrum they necessarily were equivalent.

Besides the low dimension and co-dimension cases in Theorem 3.4.2, we checked it for every $[n,k]_2$-code with $n \leqslant 11$. This was done using MAGMA. With Theorem 1.2.1 the calculation of the support of a vector space was reduced, making the algorithm work through the cases faster by doing the calculation only with the generator matrices. Lemma 3.4.1 also helped us reduce the calculations by considering the codes up to half the dimension of the ambient spaces. This because guaranteeing the conditions for the first dimensions, means it is also true for their duals or the higher dimensions. Proposition 3.4.3 helped to do some recursive work by using the results from smaller dimension to analyze higher ones. Also, by means of Theorem 3.3.1 whenever a code was equivalent to an already stored one, it was skipped and its spectrum was not calculated. All these considerations working together allowed us to compute all the equivalence classes of binary codes up to length 11 for which the task of classification by exhaustion was feasible. This gives us the next result.

Theorem 3.4.4. *Given* $C, C' \leqslant \mathbb{F}_2^n,\ n \leqslant 11$. *Then,*

$$C \equiv C' \Leftrightarrow Spec(C) = Spec(C').$$

We were not able to extend this result to general dimensions, but we do believe it should be true, so we conjecture:

Conjecture. 3.4.2. *Two binary codes are equivalent if, and only if, they have the same spectrum.*

This conjecture is known to be true in the following cases:

1. The fact that equivalent codes have the same spectrum is proved in Theorem 3.3.1.
2. The reciprocal holds for low dimension and co-dimension cases in Theorem 3.4.2.
3. It also holds for codes with equal spectrum, therefore, equal spectrum preserving basis (Theorem 3.2.2).
4. It is true for every $[n,k]_2$-codes with $n \leqslant 11$. This was checked computationally and proves Theorem 3.4.4.

It is also important to remark that the spectrum is the smallest invariant to determine the code (up to equivalence), since neither the generalized weight hierarchy nor the weight distribution can do it:

1. The weight hierarchy alone is not sufficient to guarantee the equivalence of codes, as seen in Chapter 2, Table 2 that codes with equal weight hierarchy, for instance E_{16}, F_{16}, are not equivalent.

2. The weight enumerators are also not sufficient to ensure equivalence of codes, as was shown in Example 3.0.1.

Algoritmo 1 – Algorithm to verify that equal spectrum implies equivalence

Datos: $[n, k]$
Resultado: Sequences of generator matrices and spectrums of non-equivalent $[n, k]$-codes
inicio
  $C \longleftarrow$ {All the l.i. k-tuples in $\mathbb{F}_2^n$}
  $EC, DS, CE \longleftarrow [\,]$ EC `stores Gen(C),` DS `stores Spec(C),` CE `stores counterexample if found. Initialized empty`
  para $t \in C$ **hacer**
    $cod \longleftarrow \langle t \rangle$ `the code spanned by the` t `k-tuple is obtained`
    $s \longleftarrow Spec(cod)$ `its spectrum calculated`
    si $s \notin DS$ **entonces** `If new spectrum and gen matrix stored`
      Include cod in EC
      Include s in DS
    en otro caso
      para cc *with* $Spec(cc) = s$ **hacer** `Checks the codes with equal spectrum stored`
        si $cc \not\equiv cod$ **entonces** `If not equiv. is a counterexample`
          Counterexample found, non-equivalent codes
          with same spectrum `Equivalence checked with Magma`
          $CE \longleftarrow [cc, cod]$ `This variable remained empty`
          Print CE
        fin
      fin
    fin
    Classification for $[n, k]$
    Print $[EC], [DS]$
  fin
fin

Then considering everything exposed in this chapter we have a strong belief of it to be true in general.

Doing a recap, we have seen in this chapter that even though two codes have equal enumerator polynomial, as seen in Example 3.0.1, it is not sufficient to guarantee they are equivalent codes. Also, we had already seen in Chapter 2, Table 2 that codes with equal classical parameters, for instance E_{16}, F_{16}, may have different capacity to correct errors. There, two codes of the same length, with the same dimension and same minimum distance, but nonequivalent, have different values for $\mathrm{PCD}_{\leqslant e}(C)$ and $\mathrm{PED}_{=e}(C)$ due to the differences in the distribution of subspaces' support each one has.

We have explored the concept of using the spectrum as a classification tool for binary codes. Proving that for dimension 2 it is always true that codes with equal spectrum are equivalent, and we have also seen that given some conditions it is possible to guarantee that having the same spectrum implies the codes are equivalent, the converse is always true as seen in this work also.

We have by exhaustion classified all the block codes up to length 11 3.4.4, using duality to check the dimension up to half the length 3.4.1. And in order to make the search efficient, we have used some modifications of Data Analysis techniques. First, to classify the vector spaces, searching for equivalence classes3.4.3, a modification of the K-means algorithm has been used. This is an unsupervised machine learning technique, that belongs to the clustering algorithms and in our case we used the correlation of data with respect to equivalence between the spaces. To make more efficient the searching process we have used also a variation of the Principal Component Analysis (PCA) applied to codes, that is a Dimensionality Reduction Data Analysis' algorithm by using the contrapositive of Proposition 3.3.1. It is important to remark that the machine learning algorithms were not the inspiration initially, but afterwards the link between the used technique and the basic principles of K-means and Dimensionality reduction was perceived, in fact in their pure form both are originally thought to be applied directly to classical vector spaces, but our algorithm uses the essence of each algorithm but applied to Grassmanian spaces, i.e., spaces with vector spaces as its elements. Therefore, we do not claim to have implemented the usual and pure form of K-means or Dimensionality Reduction Data Analysis but an adaptation of their spirit. In every case, as can be checked in the GitHub repository, the conjecture here presented is true up to length 11. This means that every pair of binary code of length up to 11 and any dimension are equivalent if, and only if, they have the same spectrum. Proving the remainder, in general, has been elusive due to the

conditions to guarantee that an isomorphism between subspaces of the codes can induce an isomorphism of the ambient codes. And this then gives some more opportunities to continue this research path, along with some smaller steps as to prove it for dimension 3 or higher, or getting weaker conditions to extend such an isomorphism of subcodes to the supercodes. For greater lengths this process was not computationally possible due to running time restrictions. Figure 4 shows a logarithmic graphic normalized to the running time of the case [8,4] that took about 5 min. Doing the calculations, if all the combinations are considered the case [10,5] would have taken more than 6 continuous months of CPU time and [11,5] 203, not to tell greater ones. But due to the restrictions used in the algorithm the running time for [11,6] got to about three months. This made unfeasible the pursuing of length 12.

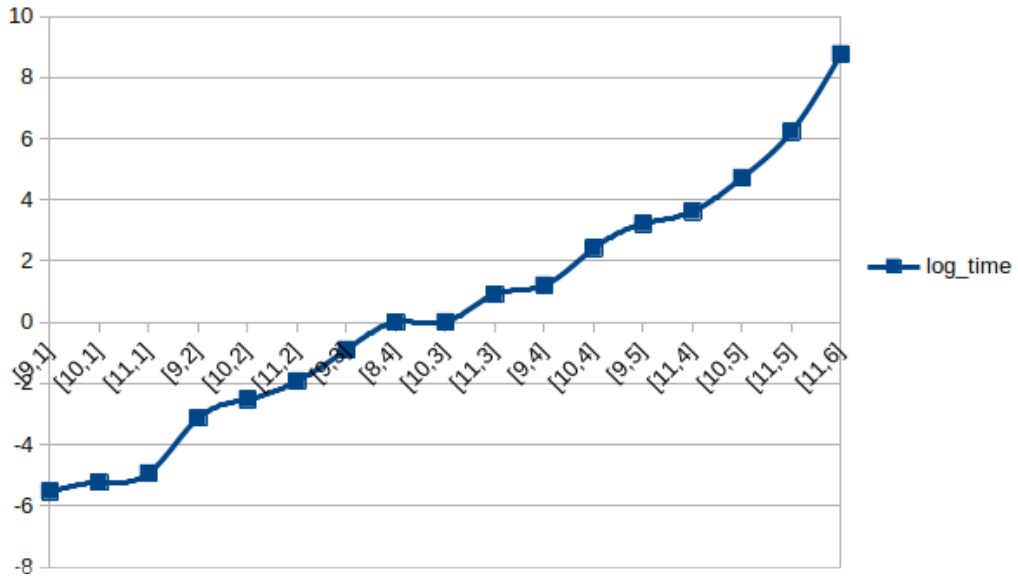


Figure 4 – CPU running time in logarithmic scale, normalized to the case [8,4]∼5 min.

3.5 Open questions related to the spectrum of codes

We have seen that the minimum distance, the weight enumerator polynomial and even the weight hierarchy are not enough to determine a code, up to isomorphism. Whereas the spectrum or the matrix of support distribution of the subcodes of C seems to be the minimal set of invariants to consider in order to guarentee equivalence of codes.

There are some results linking the concept of permutational equivalence with having the same spectrum. But we would like to extend this concept to monomial equivalence, this implies generalizing the definition of spectrum for larger fields than $\mathbb{F}_2$ and checking how the results may be adapted in those conditions. In short, we would very much like to prove the main conjecture presented in 3.4.2. Also it would be interesting to investigate the relation of this potential classification technique with cryptosystems, such as McEliece.

4 Extremal type III codes

In this chapter we work with a different problem: The classification of extremal type III codes with some fixed parameters. It is known by (SHENGYUAN, 1999) that self-dual ternary codes, much less extremal, do not exist if $n = 12i, i \geqslant 70$. That makes it interesting, investigating the possible cases yet to be completely classified. In particular, we will see that for length 60 and 52 it was possible to obtain codes that were previously unknown. By means of monomial representations we obtained a family of codes invariant under the group $\mathrm{SL}_2(p)$, including the extremal type III code of length 60. For the type III code of length 52 an algebraic construction is presented. These results have been published in (NEBE; VILLAR, 2013; BOUYUKLIEVA; CRUZ; VILLAR, 2022), and presented in ALCOMA 2015 (VILLAR, 2015a) and in CWC 2015 (VILLAR, 2015b). In order to do it, we deal first with the general concepts and necessary propositions to exclude some types of automorphisms from the list of all the possible options. Some of these due to general conditions presented and others considering the particular case to study the known bounds for the minimum distances of codes available in (GRASSL, 2007). The way to do this, is that if given $[n, k]_3$ the expected d, if forbidden in the table, then that combination may be excluded.

4.1 Definitions

Let $r \in \mathbb{N}$. We say a code C is **r-divisible**, if $r \mid \mathrm{wt}(\mathbf{c})$, for every $\mathbf{c} \in C$. Self-dual codes, those for which $C^{\perp} := \{v \in \mathbb{F}_q^n : (v \mid c) = 0, \text{ for all } c \in C\} = C$, have a special classification, depending on the field over which they are defined and their r-divisibility, as it follows: If C is a q-ary code, self-dual and r-divisible, with $r > 1$, then we say that C is a code of

a. **Type I** if $q = 2$ and C is not doubly even, i.e., $r \neq 4$.

b. **Type II** if $q = 2$ and C is doubly even.

c. **Type III** if $q = 3$, which by being self-dual is also 3-divisible.

d. **Type IV** if $q = 4$, and therefore, it is even (r=2) as well.

Gleason, et al. (ASSMUS; MATTSON; TURYN, 1967) proved that, if $s > 1$ divides the size of the support of each *codeword* in a non-trivial binary self-dual code, then either $s = 2$ or $s = 4$. Due to the self-duality, binary Type I, II codes satisfy naturally this condition, when $s = 2$. According to (GLEASON, 1971) doubly even binary self-dual codes exist if n is a multiple of eight.

In (MALLOWS; SLOANE, 1973) C.L. Mallows and N.J.A. Sloane proved that for C a ternary self-dual $[n, k, d]$-code the following inequality holds for d:

$$d \leqslant 3\left\lfloor \tfrac{n}{12} \right\rfloor + 3.$$

And in (MACWILLIAMS et al., 1978) for C a quaternary self-dual $[n, k, d]$-code it holds that:

$$d \leqslant 2\left\lfloor \tfrac{n}{6} \right\rfloor + 2,$$

here $\lfloor x \rfloor$ denotes the floor of x. Codes meeting this bound are named **extremal**. Examples of notable extremal binary codes are the $[24, 12, 8]$ extended binary Golay code and the $[8, 4, 4]$ extended Hamming code (GOLAY, 1949; HAMMING, 1950). Due to Golay it is also known the unique $[12, 6, 6]$ ternary extremal code, the extended ternary Golay code (GOLAY, 1949).

Finally, given C k-dimensional code of length n over $\mathbb{F}_q$ and $\sigma \in \mathrm{Aut}(C)$ of order p, p a prime number, we say that $\sigma \in \mathrm{Sym}(n)$ is of **type** $p-(t, f)$ if σ can be written as the composition of t p-cycles and f fixed points.

Example. 4.1.1. *Consider the permutation* $\sigma = (123)(456)(789) \in \mathrm{Sym}(14)$. *Then* σ *is of type 3-(3,5), because it is of order* $p = 3$, *there are 3 p-cycles and it has 5 fixed points in the set* $[14] = \{1, \ldots, 14\}$.

4.2 Code decomposition

When studying codes, specially large ones, it comes on handy to have the chance to break them down into pieces that represent the whole set. This because, depending on the case, the submodules turn out to be easier to handle than the complete

code. It allows us to obtain the larger code from conditioning the smaller blocks. That is what is behind the technique used in this section.

Let C be a self-dual extremal code of length n with $n = pt + f$ and we define two sets that we will see are indeed submodules of C.

Definition. 4.2.1. *Given C a self-dual extremal code, σ an automorphism of C of order p, a prime number, and type $p-(t,f)$. Suppose $\sigma = \Omega_1 \cdot \ldots \cdot \Omega_t \cdot \Omega_{t+1} \cdot \ldots \cdot \Omega_{t+f}$, where wlog we take*

$$\Omega_i := \begin{cases} (p(i-1)+1, \cdots, ip) & , \ i \in \{1, \cdots, t\} \\ (p(i-f)+f) & , i \in \{t+1, \ldots, t+f\} \end{cases}$$

Then

$$F_\sigma(C) := \left\{ \mathbf{c} \in C : \sigma(\boldsymbol{c}) = \mathbf{c} \Leftrightarrow c_1 = \cdots = c_p, \cdots, c_{p(t-1)+1} = \cdots = c_{tp} \right\},$$

is the ***fixed code*** *and*

$$E_\sigma(C) := \left\{ \mathbf{c} \in C : \sum_{i \in \Omega_1} c_i = \cdots = \sum_{i \in \Omega_t} c_i = c_{tp+1} = \cdots = c_{tp+f} = 0 \right\},$$

is the $\sigma-$ ***invariant complement of*** *$F_\sigma(C)$. We also define the functions*

$$\pi_t : \ F_\sigma(C) \longrightarrow \mathbb{F}_q^t, \qquad \pi_f : \ F_\sigma(C) \longrightarrow \mathbb{F}_q^f,$$

$$\mathbf{v} \mapsto (v_p, v_{2p}, \ldots, v_{tp}) \qquad \mathbf{v} \mapsto (v_{tp+1}, \ldots, v_{tp+f})$$

Then we can define the projection homomorphism $\pi : F_\sigma(C) \longrightarrow \mathbb{F}_q^{t+f}$ as follows.
For $\mathbf{v} \in F_\sigma(C)$,

$$\pi(\mathbf{v}) := (\pi_t(\mathbf{v}), \pi_f(\mathbf{v}))$$

The following theorem provides some information on the decomposition of a self-dual code C with an automorphism σ of order $p \nmid \mathrm{char}(\mathbb{F}_q)$. We consider the automorphism organized as in Definition 4.2.1.

Theorem. 4.2.2. *Let $C = C^\perp \leqslant \mathbb{F}_q^n$, $p \nmid \mathrm{char}(\mathbb{F}_q)$ and $\sigma \in \mathrm{Aut}(C)$ of type $p-(t,f)$ with $\sigma = \Omega_1 \cdot \ldots \cdot \Omega_t \cdot \Omega_{t+1} \cdot \ldots \cdot \Omega_{t+f}$. Then $F_\sigma(C)$ has dimension $\dfrac{f+t}{2}$ and $E_\sigma(C)$ has dimension $\dfrac{t(p-1)}{2}$.*

Proof. Using the same homomorphism π from Definition 4.2.1, we shall denote $\pi(F_\sigma(C))$ by $\overline{F_\sigma(C)}$. In fact, π defined onto $\overline{F_\sigma(C)}$ is an isomorphism. This, because from its definition one can easily check that for all $\mathbf{v}, \mathbf{w} \in F_\sigma(C)$, $\alpha \in \mathbb{F}_q$, $\pi(\mathbf{v}+\mathbf{w}) = \pi(\mathbf{v}) + \pi(\mathbf{w})$ and $\pi(\alpha\mathbf{v}) = \alpha\pi(\mathbf{v})$. Then it is a linear transformation. Now that it is surjective is clear because it is defined to its image and it is injective because if $\mathbf{v} \in F_\sigma(C)$ is such that $\pi(\mathbf{v}) = \mathbf{0}$, then again by the definition of π all the coordinates of $\mathbf{v}$ are 0 or $\mathbf{v} = \mathbf{0}$, meaning $\ker(\pi) = \{\mathbf{0}\}$. In consequence, since C is self-dual, so is $\overline{F_\sigma(C)}$, therefore $\dim(F_\sigma(C)) = \dim(\overline{F_\sigma(C)}) = \dfrac{f+t}{2}$. Thus, we get

$$\dim(E_\sigma(C)) = \frac{n}{2} - \dim(F_\sigma(C)) = \frac{tp+f}{2} - \frac{f+t}{2} = \frac{t(p-1)}{2}.$$

Now if we take π_t, since

$$F_\sigma(C)/\ker(\pi_t) \cong \mathrm{Img}(\pi_t) \leqslant \mathbb{F}_q^t,$$

we get

$$\dim(F_\sigma(C)) - \dim(\ker(\pi_t)) \leqslant t \Rightarrow \dim(\ker(\pi_t)) \geqslant \frac{f-t}{2}.$$

□

Lemma 4.2.1. *If $f < \mathrm{d}(C)$, then $t \geqslant f$.*

Proof. Suppose $f < \mathrm{d}(C)$, then $\ker(\pi_t) = \{0\}$. In fact, in general we have

$$\ker(\pi_t) = \{v \in F_\sigma(C) : v = (0, \ldots, 0, v_{t+1}, \ldots, v_{t+f})\}.$$

If $\ker(\pi_t) \neq \{0\}$ we would get a subcode with minimum distance d' smaller than d, which contradicts d being the minimum distance of the code. Hence, if $f < \mathrm{d}$, it necessarily implies $\ker(\pi_t) = \{0\}$, i.e., π_t is injective. Then

$$\dim(F_\sigma(C)) = \frac{t+f}{2} \leqslant t = \dim(\mathbb{F}_q^t) \Rightarrow f \leqslant t.$$

□

Lemma 4.2.1 is one key result to help us in the following section clean up the list of possible types of automorphisms of order p. So is Theorem 4.2.2 too.

4.3 Ternary extremal codes

In this section we work on the characterization of some ternary extremal codes, those with parameters [36,18,12], [52,26,15] and [60,30,18], all extremal. The first one corresponds to the Pless code for $p = 17$ (PLESS, 1969), the second one there exist two nonequivalent extremal Type III codes of length 52, one presented by Gaborit in (GABORIT; OTMANI, 2003) and a second one previously unknown found in this work and for the third case it was found that there exist three nonequivalent codes with such parameters, two previously known the Extended $QR(59)$ Code and the Pless code for $p = 29$ (PLESS, 1969), and a third one with an smaller automorphism group. We denote by $E_\sigma(C)^*$ the shortened code obtained from $E_\sigma(C)$ by deleting the last f coordinates. If we define $K := \ker(\pi_t)$, we also denote by K^* the subcode with support only, at most, on the last f coordinates. With the techniques used in this section only automorphisms of order $p \geqslant 5$ were studied. This since considering $p = 2, 3$ implies taking into consideration that p is even for 2 and that p divides the characteristic of the field for 3, and it requires the refinement of the statements or include others that go beyond the scope of this research.

4.3.1 Extremal Type III code of length 36

The next table shows the different options of types $p - (t, f)$ for a nontrivial automorphism of a [36,18,12]-code C. Here $36 = pt + f$.

p	2																	
t	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
f	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34

p	3												5						
t	12	11	10	9	8	7	6	5	4	3	2	1	7	6	5	4	3	2	1
f	0	3	6	9	12	15	18	21	24	27	30	33	1	6	11	16	21	26	31

p	7					11			13		17		19	23	29	31
t	5	4	3	2	1	3	2	1	2	1	2	1	1	1	1	1
f	1	8	15	22	29	3	14	25	10	23	2	19	17	13	7	5

Table 7 – List of types allowed for $n = 36$

Now we proceed to exclude from this list the types that cannot occur.

Lemma. 4.3.1. *No automorphism σ has only one p-cycle.*

Proof. If $t = 1$, then the code E^* has length p and dimension $\frac{p-1}{2}$, this means in particular that $p \geqslant 12$, and also $f \leqslant 36 - p$, i.e., $f \leqslant 23$. Then K^* is a code of dimension

at least $\frac{f-1}{2}$, length f and minimum distance $d \geqslant 12$. So, the options are for it to be a [23,11], [19,9], [17,8] or [13,6]-code, dismissing [7,3] and [5,2] for obvious reasons, but by (GRASSL, 2007) none of these are possible. □

Lemma. 4.3.2. *If $t = 2$, then the p-cycles are of length 17.*

Proof. If $t = 2$, then the code E^* has length $2p$ and dimension $p-1$ and minimum distance at least 12, this means in particular that $p \geqslant 11$ and $p \leqslant 17$. If $p = 17$, then $t = f = 2$. So let us consider $p < 17$, then either $p = 11$ or $p = 13$. Thus K^* is either a [14,6]-code or a [10,4]-code with $d \geqslant 12$, but then again by (GRASSL, 2007) none of these are possible, being the last one absurd. □

Lemma. 4.3.3. *If $t = 3$, then $p = 11$.*

Proof. If $t = 3$, then the code E^* has length $3p$, then $p \leqslant 12$ and also $d(E^*) \geqslant 12$, thus $p \geqslant 5$. If $p = 11$, then $t = f = 3$. So we analyze $p = 5,\ 7$. If such a code existed, then there would exist a [21,9] or [15,6] code with $d \geqslant 12$ and using (GRASSL, 2007) these codes do not exist. □

Lemma. 4.3.4. *If $\sigma \in Aut(C)$, then the order of $\sigma \neq 5, 7$.*

Proof. If $p = 7$ there are only two options left, the types $7-(4,8)$ and $7-(5,1)$. But if $t = 4$, $f = 8 < 12$ and $t \not\geqslant f$, thus this is not possible by remark 4.2.1. On the other hand if we have $7-(5,1)$, the kernel K of the projection of $F_\sigma(C)$ onto the first 35 coordinates is trivial, then the projection is

$$\mathbb{F}_3^5 \otimes \langle (1,1,1,1,1,1,1) \rangle.$$

This means that $(1^7, 0^{28}, f_1) \in F_\sigma(C)$, where x^n represents the vector of length n with x in all its components, which is a vector of weight at most $8 < 12$, a contradiction to the minimum distance. Then $p = 7$ cannot occur. Now for $p = 5$ we have the remaining types for which

$$t \in \{4,5,6,7\},\ \ f \in \{16,11,6,1\},$$

respectively. But if $t = 4$ then K^* would be a code of length 16 and dimension 6 with minimum distance at least 12, such code does not exist by (GRASSL, 2007). $t = 5$ is

excluded by 4.2.1. For $t = 6$ we get

$$\pi_t(F_\sigma(C)) \cong \mathbb{F}_3^6 \otimes \langle (1,1,1,1,1) \rangle$$

then $(1^5, 0^{25}, f_1, \ldots, f_6) \in F_\sigma(C)$ and is a vector of weight at most $11 < 12$, not possible. Analogously, if $t = 7$ it means $(1^5, 0^{35}, f_1) \in F_\sigma(C)$ is of weight, at most, $6 < 12$ which excludes definitely 5 as a prime dividing the order of $\mathrm{Aut}(C)$. □

Then with the techniques here presented it was possible to determine that the only possible types for an extremal Type III code of length 36 with an automorphism of order $p \geqslant 5$ out of Table 7 are **17-(2,2)** and **11-(3,3)**. Out of these two it is found that the only known Code with these parameters corresponds to the Pless Code with $p = 17$, $\mathcal{P}(17)$, for which $|\mathrm{Aut}(\mathcal{P}(17))| = 2^7 \cdot 3^2 \cdot 17 = 19584$. The type 11-(3,3) as shown in (HUFFMAN, 1992) does not yield extremal codes, since it has minimum distance 9. Then we have

Theorem 4.3.1. *Let C be an extremal Type III code of length 36 with an automorphism of order $p \geqslant 5$, then C is equivalent to the Pless code $\mathcal{P}(17)$.*

4.3.2 Extremal Type III code of length 52

In this section we consider the possible types $p - (t, f)$ for automorphisms of an extremal Type III [52,26,15]-code of C, then $52 = pt + f$.

p	2																								
t	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
f	0	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52

p	3																	5									
t	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	10	9	8	7	6	5	4	3	2	1
f	1	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	2	52	52	52	52	52	52	52	52	52

p	7							11				13				17			19		23		29	31	37	41	43	47
t	7	6	5	4	3	2	1	4	3	2	1	4	3	2	1	3	2	1	2	1	2	1	1	1	1	1	1	1
f	3	52	52	52	52	52	52	8	52	52	52	0	52	52	52	1	52	52	14	52	6	52	23	21	15	11	9	5

Table 8 – List of types allowed for $n = 52$

Lemma 4.3.2. *There is no $\sigma \in \mathrm{Aut}(C)$ with type $p - (t, f)$ and $t = 1$.*

Proof. If $t = 1$, then necessarily $p \geqslant 17$. Since Lemma 4.2.1 holds, $p = 41, 43, 47$ are excluded. For $p = 17, 19, 23, 29, 31, 37$, one gets that K^* is a code with parameters $[35, 17], [33, 16], [29, 14], [23, 12], [21, 10], [15, 7]$, respectively, and minimum distance 15, but by (GRASSL, 2007) such codes do not exist. □

Lemma 4.3.3. *There is no $\sigma \in \mathrm{Aut}(C)$ with two p-cycles.*

Proof. If $t = 2$, then $52 \geqslant 2p \geqslant 15$, then $11 \leqslant p \leqslant 23$. Thus is $p = 11, 13, 17, 19, 23$. Cases $p = 19, 23$ get excluded from Lemma 4.2.1. For the rest, it means that K^* is a code with parameters $[30, 14], [26, 12], [18, 8]$, respectively, with minimum distance 15, but again by (GRASSL, 2007) such codes do not exist. □

Lemma 4.3.4. *There is no $\sigma \in \mathrm{Aut}(C)$ with three p-cycles*

Proof. If $t = 3$, then $52 \geqslant 3p \geqslant 15$. Then $5 \leqslant p \leqslant 17$. For $p = 17$ we notice that E is an hermitian self-dual code of length 3, but it is impossible, since its length should be even. In the case $p = 13$ it contradicts Lemma 4.2.1. For the rest $p = 5, 7, 11$, we get K^* is a code with parameters $[37, 17], [31, 14], [19, 8]$, respectively, and once again checking in (GRASSL, 2007) it is not possible. □

Then we have seen there is no automorphism of C with order $p \geqslant 17$

Now we concentrate in the only option left for an autormorphism of order $p = 13$. That is, an automorphism of type 13-(4,0). With this we will verify the next proposition.

Proposition 4.3.5. *There exist at least two nonequivalent Type III codes of length 52*

In (GABORIT; OTMANI, 2003) it was introduced the first extremal Type III code of length 52 $\mathcal{G}(52)$ using constructive algorithms. In this section we focus on the only type possible of order 13, for such a code. This is the type 13-(4,0) and we find a second extremal Type III code, namely C_{52} nonequivalent to $\mathcal{G}(52)$.

Given $p,\ q$ different prime numbers. If we define

$$d := \mathrm{Ord}(q \mod p),$$

i.e., d is the smallest natural number such that $q^d \equiv 1 \bmod p$ and $a := \dfrac{p-1}{d}$, then the algebra can be decomposed as follows:

$$\mathbb{F}_q[x]/(x^p - 1) \cong \mathbb{F}_q \oplus \bigoplus_{i=1}^{a} \mathbb{F}_{q^d}.$$

For $p = 13$ and length $n = 52$ over $\mathbb{F}_3$ we may consider the Algebra

$$\mathbb{F}_3[x]/(x^{13} - 1),$$

here $(x^{13}-1) = (x-1)\cdot p_1 \cdot p_2 \cdot p_3 \cdot p_4$, $p_i,\ i = 1,2,3,4$, of degree 3, the order of 3 in $\mathbb{F}_{13}$, irreducible over $\mathbb{F}_3$. By means of the Chinese remainder theorem, we get:

$$\mathbb{F}_3[x]/(x^{13}-1) \cong \mathbb{F}_3[x]/(x-1) \oplus \mathbb{F}_3[x]/(p_1) \oplus \mathbb{F}_3[x]/(p_2) \oplus \mathbb{F}_3[x]/(p_3) \oplus \mathbb{F}_3[x]/(p_4).$$

Let e_i be the corresponding idempotent element in each submodule. In order to obtain explicit generator matrices we observe the action of $\sigma = (1\ 2\ \cdots\ 13)$ on each idempotent by evaluating each one of them in the permutation matrix of order 13×13 associated to σ, M. Let us name them $E_i := e_i(M),\ i = 0, \ldots, 4$. We take a generator matrix of the form

$$G = \left(\begin{array}{c|cc} I_{26} & \begin{matrix} \mathrm{A} & \mathrm{B} \\ \mathrm{C} & \mathrm{D} \end{matrix} \end{array} \right),$$

since we are interested on the code to be self-dual, we want

$$G \cdot G^T = 0 \Rightarrow \begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} A^T & C^T \\ B^T & D^T \end{pmatrix} = - \begin{pmatrix} I_{13} & 0 \\ 0 & I_{13} \end{pmatrix}.$$

In this case by transposing G, E_0 is fixed and $E_1,\ E_2$ and $E_3,\ E_4$ are exchanged. For instance if $A = a_0E_0 + a_1E_1 + a_2E_2 + a_3E_3 + a_4E_4$, then $A^T = a_0E_0 + a_2E_1 + a_1E_2 + a_4E_3 + a_3E_4$. Taking this into consideration we have:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} A^T & C^T \\ B^T & D^T \end{pmatrix} = \begin{pmatrix} AA^T + BB^T & AC^T + BD^T \\ CA^T + DB^T & CC^T + DD^T \end{pmatrix} = - \begin{pmatrix} I_{13} & 0 \\ 0 & I_{13} \end{pmatrix}.$$

Then we get the following equations:

- $a_0^2 + b_0^2 = -1$, $c_0^2 + d_0^2 = -1$.
- $a_0c_0 + b_0d_0 = 0$, $a_1c_2 + b_1d_2 = 0$,$a_2c_1 + b_2d_1 = 0$,$a_3c_4 + b_3d_4 = 0$,$a_4c_3 + b_4d_3 = 0$.
- $a_1a_2 + b_1b_2 = -1$, $a_3a_4 + b_3b_4 = -1$, $c_1c_2 + d_1d_2 = -1$, $c_3c_4 + d_3d_4 = -1$.

Because of the variables, we can sort the equations into the following independent systems:

1.	2.	3.
$a_0^2 + b_0^2 = -1$	$a_1c_2 + b_1d_2 = 0$	$a_3a_4 + b_3b_4 = -1$
$c_0^2 + d_0^2 = -1$.	$a_2c_1 + b_2d_1 = 0$	$a_4c_3 + b_4d_3 = 0$.
$a_0c_0 = -b_0d_0$	$a_1a_2 + b_1b_2 = -1$	$a_3a_4 + b_3b_4 = -1$
	$c_1c_2 + d_1d_2 = -1$	$c_3c_4 + d_3d_4 = -1$

Theorem. 4.3.5. *There exist 18432 double circulant self-dual codes of length 52 over F_3 with an automorphism of order $p = 13$.*

Proof. To prove this assertion we count the number of solutions of the system shown above. For the first one we know $a_0, b_0, c_0, d_0 \in \mathbb{F}_3^*$, because of the first two equations, then the third one can be rewritten as:

$$a_0 = -b_0 d_0 c_0^{-1} = -b_0 d_0 c_0.$$

This tells us that the first system has 2^3 solutions.

The second system is more complex, therefore we distinguish the possibilities. Before doing so let us remark that if $b_1 = 0$, then $a_1 a_2 = -1$, thus $a_1, a_2 \neq 0$, then $a_1 c_2 = 0 \Rightarrow c_2 = 0 \Rightarrow d_1 d_2 = -1$, then $d_1, d_2 \neq 0$, also $b_2 = 0$ lead us to $d_1, d_2 \neq 0$. Then either b_1 or b_2 equal 0, imply neither d_1 nor d_2 equal 0, and conversely. So the possible cases are:

i. $b_1 = 0,\ b_2 \neq 0.$

ii. $b_1 = 0,\ b_2 = 0.$

iii. $b_1 \neq 0,\ b_2 = 0.$

iv. $d_1 = 0,\ d_2 \neq 0.$

v. $d_1 = 0,\ d_2 = 0.$

vi. $d_1 \neq 0,\ d_2 = 0.$

vii. $b_1,\ b_2,\ d_1,\ d_2 \neq 0.$

Let's consider first the case ii. for being simpler. Here $b_1 = b_2 = 0 \Rightarrow c_1 = c_2 = 0$ and the system is then reduced to the equations $a_1 a_2 = -1,\ d_1 d_2 = -1$, equivalently $a_1 = -a_2,\ d_1 = -d_2$ in $\mathbb{F}_3$. Then we get four(4) solutions. If i., i.e., $b_1 = 0,\ b_2 \neq 0$ the system turns out to be equivalent to $a_1 a_2 = -1,\ d_1 d_2 = -1,\ a_2 c_1 = -b_2 d_1 \Leftrightarrow a_2 = -a_1,\ d_1 = -d_2,\ c_1 = -b_2 d_1 a_2 = -b_2 d_2 a_1$. But then again, $b_2, d_2, a_1 \in \mathbb{F}_3^*$, thus we get other eight(8) solutions. Analogously for iii. we get another 8 solutions and applying a similar procedure for iv., v. and vi. we get in total for the cases i. to vi. 40 solutions. So

we only have left the last case. vii.$b_1,\ b_2,\ d_1,\ d_2 \neq 0$. We have the initial system

$$a_1c_2 + b_1d_2 = 0 \tag{4.1}$$

$$a_2c_1 + b_2d_1 = 0 \tag{4.2}$$

$$a_1a_2 + b_1b_2 = -1 \tag{4.3}$$

$$c_1c_2 + d_1d_2 = -1 \tag{4.4}$$

Under these circumstances if $b_1,\ b_2,\ d_1,\ d_2 \neq 0$, then $a_1,\ a_2,\ c_1,\ c_2 \neq 0$. Then from (1) and (2) we get:

$$a_1 = -\frac{b_1d_2}{c_2}$$
$$a_2 = -\frac{b_2d_1}{c_1}$$

Also from (4) and considering that all variables are different from 0,

$$c_1c_2 = d_1d_2 \tag{4.5}$$

Using these equations in (3) we obtain:

$$\frac{b_1d_1b_2d_2}{c_1c_2} + b_1b_2 = -1$$
$$\Rightarrow \quad b_1b_2\left[\frac{d_1d_2}{c_1c_2} + 1\right] = -1$$
$$\Rightarrow \quad b_1b_2 = -\left[\frac{d_1d_2}{c_1c_2} + 1\right] = -2 = 1$$
$$\Rightarrow \quad b_1b_2 = 1$$
$$\Rightarrow \quad b_1 = b_2$$

With this result used in equation (3):

$$a_1a_2 + b_1^2 = -1$$
$$\Rightarrow \quad a_1a_2 + 1 = -1$$
$$\Rightarrow \quad a_1a_2 = 1$$
$$\Rightarrow \quad a_1 = a_2$$

And from (5) and (4) we get $2d_1d_2 = -1 \Rightarrow d_1d_2 = 1 \Rightarrow d_1 = d_2$ and $c_1 = c_2$, thus our system turns out to be equivalent to:

$$a_1 = a_2 = -b_1d_1c_1$$
$$d_2 = d_1$$
$$b_2 = b_1$$
$$c_2 = c_1,$$

where $b_1, c_1, d_1 \in \mathbb{F}_3^*$, then we get eight additional solutions. In total, we get for our second system 48 different solutions. Finally, for our third system if we identify

$$a_1 \leftrightarrow a_3b_1 \leftrightarrow b_3 \qquad c_1 \leftrightarrow c_3d_1 \leftrightarrow d_3$$
$$a_2 \leftrightarrow a_4b_2 \leftrightarrow b_4 \qquad c_2 \leftrightarrow c_4d_2 \leftrightarrow d_4$$

We notice that the second and third system are equivalent, therefore this one also has 48 solutions. This means that we get $8 \cdot 48 \cdot 48 = 18432$ doubly circulant self-dual codes out of which the calculations with MAGMA say 384 are extremal and because of the construction also equivalent to the new $[52, 26, 15]_3-$ code found in this research project.

□

We checked with MAGMA the automorphism group of both $\mathcal{G}(52)$ and C_{52}. And we get that

- $|\operatorname{Aut}(\mathcal{G}(52)| = 2^5 \cdot 13 = 416$.
- $|\operatorname{Aut}(C_{52})| = 2^2 \cdot 3 \cdot 13 = 156$.

Where we can see clearly both are nonequivalent. And also that trying to generalize the construction of C_{52} is not easy because its automorphism group is pretty small and has no rich structure.

4.3.3 Extremal Type III code of length 60

Analogously to the previous cases, Table 9 shows the different options of types to a nontrivial automorphism of an extremal Type III [60,30,18]-code C. Here $60 = pt + f$.

Lemma. 4.3.6. *If $t = 1$, then $p = 59$.*

p	2																		
t	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12
f	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36
p	2											3							
t	11	10	9	8	7	6	5	4	3	2	1	20	19	18	17	16	15	14	13
f	38	40	42	44	46	48	50	52	54	56	58	0	3	6	9	12	15	18	21
p	3												5						
t	12	11	10	9	8	7	6	5	4	3	2	1	12	11	10	9	8	7	6
f	24	27	30	33	36	39	42	45	48	51	54	57	0	5	10	15	20	25	30
p	5					7							11						13
t	5	4	3	2	1	8	7	6	5	4	3	2	1	5	4	3	2	1	4
f	35	40	45	50	55	4	11	18	25	32	39	46	53	5	16	27	38	49	8

p	13			17			19			23		29		31	37	41	43	47	53	59
t	3	2	1	3	2	1	3	2	1	2	1	2	1	1	1	1	1	1	1	1
f	21	34	47	9	26	43	3	22	41	14	37	2	31	29	23	19	17	13	7	1

Table 9 – List of types allowed for $n = 60$

Proof. Since $E_\sigma(C)^*$ is of length p and of minimum distance d $\geqslant 18$, we get

$$p \in \{19, 23, 29, 31, 37, 41, 43, 47, 53, 59\}.$$

If $p = 59$, $t = f = 1$. So we consider $p < 59$, here K^* is of length f and minimum distance at least 18. But the codes, for every p, [41,20], [37,18], [31,15], [29,14], [23,11], [19,9], [17,8], [13,6] and [7,3] do not exist by (GRASSL, 2007), being the last three on first sight impossible. □

Lemma. 4.3.7. *If $t = 2$, then $p = 29$.*

Proof. Since $E_\sigma(C)^*$ is of length $2p$ and of minimum distance d $\geqslant 18$, we get

$$p \in \{11, 13, 17, 19, 23, 29\} \text{ and } f \in \{38, 34, 26, 22, 14, 2\},$$

respectively. If $p = 29$, $t = f = 2$. So we consider $p < 29$, here K^* is of length f and minimum distance at least 18. But the codes, for every p, [38,18], [34,16], [26,12], [22,10] and [14,6] do not exist by (GRASSL, 2007), being the last one immediately impossible. □

This last lemma tells us that if an extremal Type III code with parameters [60,30,18] has an automorphism of order 29, then it is of the type 29-(2;2). Our purpose now is to prove the following theorem.

Theorem 4.3.6. *There are exactly three nonequivalent extremal [60,30,18] Type III codes with an automorphism of order $p = 29$.*

Proof. For $\sigma \in \mathrm{Aut}(C)$ of order $p = 29$ we have that $E_\sigma(C)$ is a one-dimensional isotropic subspace of $\mathbb{F}_{3^{28}}^2$, isotropic with respect to some unitary form. In (TAYLOR, 1992) we have that for V a unitary geometry of dimension n over $\mathbb{F}_{q^2}$, then the number of totally isotropic k-dimensional subspaces in V is

$$\prod_{i=n+1-2k}^{n} (q^i - (-1)^i) \Big/ \prod_{j=1}^{k} (q^{2j} - 1).$$

Then for our scenario we have $q = 3^{14},\ n = 2,\ k = 1$ and we get:

$$\prod_{i=2+1-2(1)}^{2} ((3^{14})^i - (-1)^i) \Big/ \prod_{j=1}^{1} ((3^{14})^{2j} - 1) = (3^{14} + 1) = 4782970.$$

This means, there are 4782970 possibilities for $E_\sigma(C)$. The group $G := \mathrm{SU}(2, 3^{14})$ acts transitively on these subspaces, so they can be computed as an orbit. In this scenario

$$E_\sigma(C) = \left\{ (a, b, 0, 0) \middle| a, b \in \mathbb{F}_3^{29}, \sum_i a_i = \sum_i b_i \equiv 0 \mod 3 \right\} \cong \mathbb{F}_{3^{28}}^2.$$

Then these elements in $\mathbb{F}_{3^{28}}^2$ are identified with words in $\mathbb{F}_3^{60}$ and we obtain generator matrices for $E_\sigma(C)$ of the form $(* \mid \mathbf{0}\ \mathbf{0}) \in \mathbb{F}_q^{28\times 60}$, here $\mathbf{0} \in \mathbb{F}_q^{28\times 1}$. Also as in Theorem 4.2.2 we have that $\dim(F_\sigma(C)) = \dfrac{t+f}{2} = 2$ and this leaves us $F_\sigma(C)$ uniquely determined up to equivalence as follows:

$$\mathrm{gen}(F_\sigma(C)) = \left(\begin{array}{cc|cc} \mathbf{1} & \mathbf{0} & 1 & 0 \\ \mathbf{0} & \mathbf{1} & 0 & 1 \end{array} \right),$$

where $\mathbf{1}$ is the all-one vector and $\mathbf{0}$ the zero-vector of length 29. Doing exhaustively this process we found three inequivalent codes, the extended quadratic residue code for $p = 59$ or XQR(59), the Pless code for $p = 29$ or $\mathcal{P}(29)$, and a new extremal Type III code for $p = 29$ named $\mathcal{V}(29)$. They have the following generator matrices and using MAGMA one gets the automorphism groups:

- $\mathrm{gen}(\mathrm{XQR}(59)) = (I_{30} \mid A_1)$, $\mathrm{Aut}(\mathrm{XQR}(59)) = \mathrm{SL}_2(59) \times C_2$
  $|\mathrm{Aut}(\mathrm{XQR}(59))| = 205320 = 2^3 \cdot 3 \cdot 5 \cdot 29 \cdot 59$
- $\mathrm{gen}(\mathcal{P}(29)) = (I_{30} \mid A_2)$, $\mathrm{Aut}(\mathcal{P}(29)) = (\mathrm{PSL}_2(59) \times C_4) \cdot 2$
  $|\mathrm{Aut}(\mathcal{P}(29))| = 97440 = 2^5 \cdot 3 \cdot 5 \cdot 7 \cdot 29$
- $\mathrm{gen}(\mathcal{V}(29)) = (I_{30} \mid A_3)$, $\mathrm{Aut}(\mathcal{V}(29)) = \mathrm{SL}_2(29) \times C_2$
  $|\mathrm{Aut}(\mathcal{V}(29))| = 24360 = 2^3 \cdot 3 \cdot 5 \cdot 7 \cdot 29$

Note the codes are clearly inequivalent since the order of their automorphism group are all different, reinforcing the result already obtained from the computations with MAGMA. Matrices $A_i,\ i = 1, 2, 3$ are:

$$A_1 = \begin{pmatrix}
2\,1\,1\,2\,0\,1\,1\,0\,2\,1\,0\,0\,1\,1\,0\,1\,0\,1\,2\,2\,2\,2\,1\,1\,0\,2\,0\,0\,2\,0\\
0\,2\,1\,1\,2\,0\,1\,1\,0\,2\,1\,0\,0\,1\,1\,0\,1\,0\,1\,2\,2\,2\,2\,1\,1\,0\,2\,0\,2\,0\\
0\,0\,2\,1\,1\,2\,0\,1\,1\,0\,2\,1\,0\,0\,1\,1\,0\,1\,0\,1\,2\,2\,2\,2\,1\,1\,0\,2\,2\,0\\
1\,1\,1\,0\,2\,2\,0\,1\,2\,2\,1\,0\,2\,1\,1\,2\,2\,1\,2\,1\,2\,0\,0\,0\,0\,2\,2\,1\,2\,1\\
2\,0\,0\,0\,2\,1\,1\,2\,0\,1\,1\,0\,2\,1\,0\,0\,1\,1\,0\,1\,0\,1\,2\,2\,2\,2\,1\,1\,2\,0\\
2\,1\,2\,2\,2\,1\,0\,0\,1\,2\,0\,0\,2\,1\,0\,2\,2\,0\,0\,2\,0\,2\,0\,1\,1\,1\,1\,0\,2\,2\\
0\,2\,1\,2\,2\,2\,1\,0\,0\,1\,2\,0\,0\,2\,1\,0\,2\,2\,0\,0\,2\,0\,2\,0\,1\,1\,1\,1\,2\,2\\
2\,2\,1\,0\,1\,1\,1\,0\,2\,2\,0\,1\,2\,2\,1\,0\,2\,1\,1\,2\,2\,1\,2\,1\,2\,0\,0\,0\,2\,1\\
0\,2\,2\,1\,0\,1\,1\,1\,0\,2\,2\,0\,1\,2\,2\,1\,0\,2\,1\,1\,2\,2\,1\,2\,1\,2\,0\,0\,2\,1\\
0\,0\,2\,2\,1\,0\,1\,1\,1\,0\,2\,2\,0\,1\,2\,2\,1\,0\,2\,1\,1\,2\,2\,1\,2\,1\,2\,0\,2\,1\\
0\,0\,0\,2\,2\,1\,0\,1\,1\,1\,0\,2\,2\,0\,1\,2\,2\,1\,0\,2\,1\,1\,2\,2\,1\,2\,1\,2\,2\,1\\
1\,1\,1\,1\,0\,0\,2\,1\,2\,2\,2\,1\,0\,0\,1\,2\,0\,0\,2\,1\,0\,2\,2\,0\,0\,2\,0\,2\,2\,2\\
1\,2\,2\,2\,2\,1\,1\,0\,2\,0\,0\,0\,2\,1\,1\,2\,0\,1\,1\,0\,2\,1\,0\,0\,1\,1\,0\,1\,2\,0\\
2\,0\,1\,1\,1\,1\,0\,0\,2\,1\,2\,2\,2\,1\,0\,0\,1\,2\,0\,0\,2\,1\,0\,2\,2\,0\,0\,2\,2\,2\\
1\,0\,1\,2\,2\,2\,2\,1\,1\,0\,2\,0\,0\,0\,2\,1\,1\,2\,0\,1\,1\,0\,2\,1\,0\,0\,1\,1\,2\,0\\
2\,0\,2\,0\,1\,1\,1\,1\,0\,0\,2\,1\,2\,2\,2\,1\,0\,0\,1\,2\,0\,0\,2\,1\,0\,2\,2\,0\,2\,2\\
0\,2\,0\,2\,0\,1\,1\,1\,1\,0\,0\,2\,1\,2\,2\,2\,1\,0\,0\,1\,2\,0\,0\,2\,1\,0\,2\,2\,2\,2\\
1\,1\,0\,1\,0\,1\,2\,2\,2\,2\,1\,1\,0\,2\,0\,0\,0\,2\,1\,1\,2\,0\,1\,1\,0\,2\,1\,0\,2\,0\\
0\,1\,1\,0\,1\,0\,1\,2\,2\,2\,2\,1\,1\,0\,2\,0\,0\,0\,2\,1\,1\,2\,0\,1\,1\,0\,2\,1\,2\,0\\
2\,2\,0\,0\,2\,0\,2\,0\,1\,1\,1\,1\,0\,0\,2\,1\,2\,2\,2\,1\,0\,0\,1\,2\,0\,0\,2\,1\,2\,2\\
2\,1\,1\,2\,2\,1\,2\,1\,2\,0\,0\,0\,0\,2\,2\,1\,0\,1\,1\,1\,0\,2\,2\,0\,1\,2\,2\,1\,2\,1\\
2\,1\,0\,0\,1\,1\,0\,1\,0\,1\,2\,2\,2\,2\,1\,1\,0\,2\,0\,0\,0\,2\,1\,1\,2\,0\,1\,1\,2\,0\\
2\,1\,0\,2\,2\,0\,0\,2\,0\,2\,0\,1\,1\,1\,1\,0\,0\,2\,1\,2\,2\,2\,1\,0\,0\,1\,2\,0\,2\,2\\
0\,2\,1\,0\,2\,2\,0\,0\,2\,0\,2\,0\,1\,1\,1\,1\,0\,0\,2\,1\,2\,2\,2\,1\,0\,0\,1\,2\,2\,2\\
1\,1\,0\,2\,1\,0\,0\,1\,1\,0\,1\,0\,1\,2\,2\,2\,2\,1\,1\,0\,2\,0\,0\,0\,2\,1\,1\,2\,2\,0\\
1\,2\,2\,1\,0\,2\,1\,1\,2\,2\,1\,2\,1\,2\,0\,0\,0\,0\,2\,2\,1\,0\,1\,1\,1\,0\,2\,2\,2\,1\\
1\,2\,0\,0\,2\,1\,0\,2\,2\,0\,0\,2\,0\,2\,0\,1\,1\,1\,1\,0\,0\,2\,1\,2\,2\,2\,1\,0\,2\,2\\
0\,1\,2\,0\,0\,2\,1\,0\,2\,2\,0\,0\,2\,0\,2\,0\,1\,1\,1\,1\,0\,0\,2\,1\,2\,2\,2\,1\,2\,2\\
2\,2\,0\,1\,2\,2\,1\,0\,2\,1\,1\,2\,2\,1\,2\,1\,2\,0\,0\,0\,0\,2\,2\,1\,0\,1\,1\,1\,2\,1\\
1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,0\,1
\end{pmatrix}$$

$$A_2 = \begin{pmatrix}
1&1&0&0&0&0&1&0&1&1&1&0&1&1&0&1&1&1&0&1&0&0&0&0&1&1&0&2&2&2\\
1&2&2&1&1&1&1&2&1&2&2&2&1&2&2&1&2&2&2&1&2&1&1&1&1&2&2&1&2&0\\
2&0&1&1&0&0&0&0&1&0&1&1&1&0&1&1&0&1&1&1&0&1&0&0&0&0&1&1&2&2\\
2&1&2&0&0&2&2&2&2&0&2&0&0&0&2&0&0&2&0&0&0&2&0&2&2&2&2&0&2&1\\
0&2&1&2&0&0&2&2&2&2&0&2&0&0&0&2&0&0&2&0&0&0&2&0&2&2&2&2&2&1\\
1&1&0&2&0&1&1&0&0&0&0&1&0&1&1&1&0&1&1&0&1&1&1&0&1&0&0&0&2&2\\
0&1&1&0&2&0&1&1&0&0&0&0&1&0&1&1&1&0&1&1&0&1&1&1&0&1&0&0&2&2\\
0&0&1&1&0&2&0&1&1&0&0&0&0&1&0&1&1&1&0&1&1&0&1&1&1&0&1&0&2&2\\
0&0&0&1&1&0&2&0&1&1&0&0&0&0&1&0&1&1&1&0&1&1&0&1&1&1&0&1&2&2\\
2&2&2&2&0&0&2&1&2&0&0&2&2&2&2&0&2&0&0&0&2&0&0&2&0&0&0&2&2&1\\
1&0&0&0&0&1&1&0&2&0&1&1&0&0&0&0&1&0&1&1&1&0&1&1&0&1&1&1&2&2\\
2&0&2&2&2&2&0&0&2&1&2&0&0&2&2&2&2&0&2&0&0&0&2&0&0&2&0&0&2&1\\
0&2&0&2&2&2&2&0&0&2&1&2&0&0&2&2&2&2&0&2&0&0&0&2&0&0&2&0&2&1\\
0&0&2&0&2&2&2&2&0&0&2&1&2&0&0&2&2&2&2&0&2&0&0&0&2&0&0&2&2&1\\
1&1&1&0&1&0&0&0&0&1&1&0&2&0&1&1&0&0&0&0&1&0&1&1&1&0&1&1&2&2\\
2&0&0&0&2&0&2&2&2&2&0&0&2&1&2&0&0&2&2&2&2&0&2&0&0&0&2&0&2&1\\
0&2&0&0&0&2&0&2&2&2&2&0&0&2&1&2&0&0&2&2&2&2&0&2&0&0&0&2&2&1\\
1&1&0&1&1&1&0&1&0&0&0&0&1&1&0&2&0&1&1&0&0&0&0&1&0&1&1&1&2&2\\
2&0&0&2&0&0&0&2&0&2&2&2&2&0&0&2&1&2&0&0&2&2&2&2&0&2&0&0&2&1\\
0&2&0&0&2&0&0&0&2&0&2&2&2&2&0&0&2&1&2&0&0&2&2&2&2&0&2&0&2&1\\
0&0&2&0&0&2&0&0&0&2&0&2&2&2&2&0&0&2&1&2&0&0&2&2&2&2&0&2&2&1\\
1&1&1&0&1&1&0&1&1&1&0&1&0&0&0&0&1&1&0&2&0&1&1&0&0&0&0&1&2&2\\
2&0&0&0&2&0&0&2&0&0&0&2&0&2&2&2&2&0&0&2&1&2&0&0&2&2&2&2&2&1\\
1&0&1&1&1&0&1&1&0&1&1&1&0&1&0&0&0&0&1&1&0&2&0&1&1&0&0&0&2&2\\
0&1&0&1&1&1&0&1&1&0&1&1&1&0&1&0&0&0&0&1&1&0&2&0&1&1&0&0&2&2\\
0&0&1&0&1&1&1&0&1&1&0&1&1&1&0&1&0&0&0&0&1&1&0&2&0&1&1&0&2&2\\
0&0&0&1&0&1&1&1&0&1&1&0&1&1&1&0&1&0&0&0&0&1&1&0&2&0&1&1&2&2\\
2&2&2&2&0&2&0&0&0&2&0&0&2&0&0&0&2&0&2&2&2&2&0&0&2&1&2&0&2&1\\
0&2&2&2&2&0&2&0&0&0&2&0&0&2&0&0&0&2&0&2&2&2&2&0&0&2&1&2&2&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1
\end{pmatrix}$$

$$A_3 = \begin{pmatrix}
1&2&0&0&0&2&2&2&1&0&1&1&2&2&2&0&2&2&2&1&2&2&1&2&0&1&2&1&2&0\\
2&0&1&2&2&2&1&1&1&0&2&0&0&1&1&1&2&1&1&1&0&1&1&0&1&2&0&1&2&2\\
2&1&2&0&1&1&1&0&0&0&2&1&2&2&0&0&0&1&0&0&0&2&0&0&2&0&1&2&2&1\\
1&0&2&0&1&2&2&2&1&1&1&0&2&0&0&1&1&1&2&1&1&1&0&1&1&0&1&2&2&2\\
1&2&1&0&1&2&0&0&0&2&2&2&1&0&1&1&2&2&2&0&2&2&2&1&2&2&1&2&2&0\\
1&2&0&2&1&2&0&1&1&1&0&0&0&2&1&2&2&0&0&0&1&0&0&0&2&0&0&2&2&1\\
1&2&0&1&0&2&0&1&2&2&2&1&1&1&0&2&0&0&1&1&1&2&1&1&1&0&1&1&2&2\\
2&0&1&2&0&2&1&2&0&1&1&1&0&0&0&2&1&2&2&0&0&0&1&0&0&0&2&0&2&1\\
0&2&0&1&2&0&2&1&2&0&1&1&1&0&0&0&2&1&2&2&0&0&0&1&0&0&0&2&2&1\\
1&1&0&1&2&0&1&0&2&0&1&2&2&2&1&1&1&0&2&0&0&1&1&1&2&1&1&1&2&2\\
2&0&0&2&0&1&2&0&2&1&2&0&1&1&1&0&0&0&2&1&2&2&0&0&0&1&0&0&2&1\\
0&2&0&0&2&0&1&2&0&2&1&2&0&1&1&1&0&0&0&2&1&2&2&0&0&0&1&0&2&1\\
0&0&2&0&0&2&0&1&2&0&2&1&2&0&1&1&1&0&0&0&2&1&2&2&0&0&0&1&2&1\\
2&2&2&1&2&2&1&2&0&1&2&1&0&1&2&0&0&0&2&2&2&1&0&1&1&2&2&2&2&0\\
1&0&0&0&2&0&0&2&0&1&2&0&2&1&2&0&1&1&1&0&0&0&2&1&2&2&0&0&2&1\\
0&1&0&0&0&2&0&0&2&0&1&2&0&2&1&2&0&1&1&1&0&0&0&2&1&2&2&0&2&1\\
0&0&1&0&0&0&2&0&0&2&0&1&2&0&2&1&2&0&1&1&1&0&0&0&2&1&2&2&2&1\\
1&1&1&2&1&1&1&0&1&1&0&1&2&0&1&0&2&0&1&2&2&2&1&1&1&0&2&0&2&2\\
0&1&1&1&2&1&1&1&0&1&1&0&1&2&0&1&0&2&0&1&2&2&2&1&1&1&0&2&2&2\\
1&1&2&2&2&0&2&2&2&1&2&2&1&2&0&1&2&1&0&1&2&0&0&0&2&2&2&1&2&0\\
2&0&0&1&1&1&2&1&1&1&0&1&1&0&1&2&0&1&0&2&0&1&2&2&2&1&1&1&2&2\\
2&1&2&2&0&0&0&1&0&0&0&2&0&0&2&0&1&2&0&2&1&2&0&1&1&1&0&0&2&1\\
0&2&1&2&2&0&0&0&1&0&0&0&2&0&0&2&0&1&2&0&2&1&2&0&1&1&1&0&2&1\\
0&0&2&1&2&2&0&0&0&1&0&0&0&2&0&0&2&0&1&2&0&2&1&2&0&1&1&1&2&1\\
2&2&2&1&0&1&1&2&2&2&0&2&2&2&1&2&2&1&2&0&1&2&1&0&1&2&0&0&2&0\\
0&2&2&2&1&0&1&1&2&2&2&0&2&2&2&1&2&2&1&2&0&1&2&1&0&1&2&0&2&0\\
0&0&2&2&2&1&0&1&1&2&2&2&0&2&2&2&1&2&2&1&2&0&1&2&1&0&1&2&2&0\\
1&1&1&0&0&0&2&1&2&2&0&0&0&1&0&0&0&2&0&0&2&0&1&2&0&2&1&2&2&1\\
1&2&2&2&1&1&1&0&2&0&0&1&1&1&2&1&1&1&0&1&1&0&1&2&0&1&0&2&2&2\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&
\end{pmatrix}$$

□

4.3.3.1 Alternative method to classify [60,30] extremal codes over $\mathbb{F}_2$ and $\mathbb{F}_3$

The contents of this subsection is the result of a joint effort with professors Javier de la Cruz and Stefka Bouyuklieva and it takes part of a paper that has been published in the Journal Mathematics(BOUYUKLIEVA; CRUZ; VILLAR, 2022).

The most general definition for equivalence of linear codes of length n over the finite field $\mathbb{F}_q$ is based on the action of the semilinear isometries group $\mathcal{M}_n^*(q) = \mathrm{Mon}_n(\mathbb{F}_q^*) \rtimes \mathrm{Aut}(\mathbb{F}_q) \leqslant \Gamma_n(\mathbb{F}_q)$ on the vector space $\mathbb{F}_q^n$, where $\Gamma_n(\mathbb{F}_q)$ is the set of all semilinear mappings, i.e. the general semilinear group, $\mathrm{Mon}_n(\mathbb{F}_q^*)$ is the group of all monomial $n \times n$ matrices over $\mathbb{F}_q$, and $\mathrm{Aut}(\mathbb{F}_q)$ is the automorphisms group of the field $\mathbb{F}_q$. Linear q-ary codes C and C' of the same length n are equivalent whenever $C' = CT$ for some

$T \in \mathcal{M}_n^*(q)$. If $CT = C$ for an element $T \in \mathcal{M}_n^*(q)$ then T is called an automorphism of the code. The set of all automorphisms of C form a group denoted by $\mathrm{Aut}(C)$.

Any element $T \in \mathcal{M}_n^*(q)$ can be written as $T = PD\tau$ where P is a permutation matrix (permutation part), D is a diagonal regular matrix (diagonal part), and $\tau \in \mathrm{Aut}(\mathbb{F}_q)$. Note that in the case of prime q, $\mathcal{M}_n^*(q) = \mathrm{Mon}_n(\mathbb{F}_q^*)$, and if $q = 2$ then $\mathcal{M}_n^*(q) \cong \mathrm{Sym}(n)$ where $\mathrm{Sym}(n)$ is the symmetric group of degree n. The following lemma implies that in some cases, when considering automorphisms of prime order, we only need to examine permutation automorphisms.

Lemma 4.3.7. *(HUFFMAN, 1992) Let C be a linear code over $\mathbb{F}_q$ with an automorphism $T = PD\tau$ of prime order r where $r \nmid (q-1)$ and $r \nmid |\mathrm{Gal}(\mathbb{F}_q)|$. Then there exists a code C' equivalent to C where $P \in \mathrm{Aut}(C')$.*

We consider codes over $\mathbb{F}_2$ and $\mathbb{F}_3$ having an automorphism of odd prime order r. For these fields r satisfies the conditions from Lemma 4.3.7 and therefore we can use only permutation automorphisms of order r. So instead the action of the group $\mathcal{M}_n^*(q)$, we use the action of the symmetric group $\mathrm{Sym}(n)$ on $\mathbb{F}_q^n$ defined by $v\sigma := (v_{\sigma^{-1}(1)}, \ldots, v_{\sigma^{-1}(n)})$, where $v = (v_1, \ldots, v_n) \in \mathbb{F}_2^n$ and $\sigma \in \mathrm{Sym}(n)$.

To apply this theory in a general context a reinterpretation of Theorem 4.2.2 is presented.

Let $C \leqslant \mathbb{F}_q^n$ be a code with a permutation automorphism $\sigma \in \mathrm{Sym}(n)$ of order r with c cycles of length r and f fixed points. In this case, we say that σ is of type r-(c, f). Without loss of generality we can assume that

$$\sigma = \Omega_1 \ldots \Omega_c \Omega_{c+1} \ldots \Omega_{c+f} \tag{4.6}$$

where $\Omega_i = ((i-1)r + 1, \ldots, ir), i = 1, \ldots, c$, are the cycles of length r, and $\Omega_{c+i} = (cr + i), i = 1, \ldots, f$, are the fixed points. Obviously, $cr + f = n$.

We put

$$F_\sigma(C) := \{v \in C : v\sigma = v\}$$

and

$$E_\sigma(C) := \{v \in C : \sum_{i \in \Omega_j} v_i = 0 \text{ for all } j = 1, \ldots, c + f\}.$$

We use the Euclidean inner product over the field $\mathbb{F}_q$, namely

$$u \cdot v = \sum_{i=1}^{n} u_i v_i, \quad u = (u_1, \ldots, u_n), v = (v_1, \ldots, v_n) \in \mathbb{F}_q^n. \tag{4.7}$$

The following theorem gives a very important decomposition of the linear code C.

Theorem 4.3.8. *(HUFFMAN, 1986) Let $C \leqslant \mathbb{F}_q^n$ be a code with a permutation automorphism $\sigma \in Sym(n)$ of order r such that $char(\mathbb{F}_q) \nmid r$. Then the following hold.*

(i) $C = F_\sigma(C) \oplus E_\sigma(C)$. Both $F_\sigma(C)$ and $E_\sigma(C)$ are σ-invariant.

(ii) If C is self-dual under (4.7), then $\dim(F_\sigma(C)) = (c+f)/2$ and $\dim(E_\sigma(C)) = c(r-1)/2$.

Note that $v \in F_\sigma(C)$ if and only if $v \in C$ and $v\,|_{\Omega_j}$ is constant for $j = 1, \ldots, c+f$. Therefore, the map $\pi : F_\sigma(C) \to \mathbb{F}_q^{c+f}$ define by $(\pi(v))_j = v_i$ for some $i \in \Omega_j$, $j = 1, 2, \ldots, c+f$, $v \in F_\sigma(C)$, is a monomorphism.

Theorem 4.3.9. *(HUFFMAN, 1986; NEBE, 2012) Assume C is a self-dual $[\mathrm{n}, \mathrm{n}/2, \mathrm{d}]_q$ code under the inner product (4.7), and $p = char(\mathbb{F}_q)$. Then $C_\pi = \pi(F_\sigma(C))$ is a $[c+f, (c+f)/2, d_\pi]_q$ self-dual code with respect to the inner product*

$$u \cdot v = \sum_{i=1}^{c} r u_i v_i + \sum_{i=c+1}^{c+f} u_i v_i. \tag{4.8}$$

If either $r \equiv 1 \pmod p$ or $f = 0$, C_π is self-dual under (4.7).

For the remainder of this section, we assume that σ is a permutation automorphism of C of prime order $r = p$ different from $\mathrm{char}(\mathbb{F}_q)$. By $s(q,p)$ we denote the multiplicative order of q modulo p, $s(q,p) = \mathrm{ord}_p(q)$. In this work, we focus on the case when $s(q,p) = p-1$. Then the polynomial $1 + x + \cdots + x^{p-1}$ is irreducible over the field $\mathbb{F}_q$. Let $\mathcal{P}$ be the principal ideal of $\mathcal{R}_p = \mathbb{F}_q[x]/(x^p - 1)$ generated by the polynomial $(1-x)$. Obviously, $\mathcal{P} = \{v(x) \in \mathcal{R}_p : \sum_{i=0}^{p-1} v_i = 0\}$. The following result generalizes Lemma 4 of (HUFFMAN, 1982).

Lemma 4.3.10. *(HUFFMAN, 1986) If $1 + x + x^2 + \cdots + x^{p-1}$ is irreducible over $\mathbb{F}_q$, then $\mathcal{P}$ is a finite field with q^{p-1} elements. The identity is $(-1/p)((1-p) + x + x^2 + \cdots + x^{p-1}$.*

Multiplication by $(-1/p)(1+(1-p)x+x^2+\cdots+x^{p-1})$ *in* $\mathcal{P}$ *corresponds to multiplication by* $x \mod (x^p-1)$.

Let $E_\sigma(C)^*$ denote the code $E_\sigma(C)$ without the last f coordinates. For $v \in E_\sigma(C)^*$ we identify $v\,|_{\Omega_j} = (v_0, v_1, \ldots, v_{p-1})$ with the polynomial $v_0 + v_1 x + \cdots + v_{p-1}x^{p-1}$ from $\mathcal{P} \subset \mathcal{R}_p$. Thus, we obtain the map $\varphi : E_\sigma(C)^* \longrightarrow P^c$. Results in (HUFFMAN, 1982) and (YORGOV, 1983) show that if $q = 2$ and p is prime, $C_\varphi = \varphi(E_\sigma(C)^*)$ is self-dual with respect to a given inner product. Huffman generalized this in the following theorem.

Theorem 4.3.11. *(HUFFMAN, 1986) Assume that* C *is a self-dual* $[n, n/2, d]$ *code under* (4.7) *and that* $1 + x + x^2 + \cdots + x^{p-1}$ *is irreducible over* $\mathbb{F}_q$. *Suppose that there is a nonnegative integer* t *such that* $q^t \equiv -1 \pmod p$. *Then* C_φ *is a* $[c, c/2, d']$ *self-dual code over* $\mathcal{P}$ *under the inner product* $\langle \cdot, \cdot \rangle$ *given by*

$$\langle u, v \rangle = \sum_{i=1}^{c} u_i v_i^{q^t}, \tag{4.9}$$

where $u = (u_1, \ldots, u_c)$, $v = (v_1, \ldots, v_c) \in \mathcal{P}^c$.

On $\mathcal{P}^c$, we can use the Hermitian inner product, defined in (LING; SOLÉ, 2001): for $u = (u_1, \ldots, u_c)$ and $v = (v_1, \ldots, u_c)$

$$u \cdot v = \sum_{i=1}^{c} u_i \overline{v_i}, \tag{4.10}$$

where $\overline{v_i} = v_i(x^{-1}) = v_i(x^{p-1})$.

Remark 4.3.12. *In the last theorem note that* $v_i(x^{-1}) = v_i(x^{q^t}) = v_i(x)^{q^t}$. *Therefore, the Hermitian product (4.10) is equivalent to*

$$u \cdot v = \sum_{i=1}^{c} u_i v_i^{q^t}.$$

Moreover, if $s(q,p) = p-1$ *and* $p \neq 2$, *then* $q^{\frac{p-1}{2}} \equiv -1 \mod p$. *Therefore, we can take* $t = \dfrac{p-1}{2}$.

The following theorem is an immediate generalization of (YORGOV, 1983, Theorem 3).

Theorem 4.3.13. *Let $s(q,p) = p-1$ and $C \leqslant \mathbb{F}_q^n$ is a linear code. Suppose that there is a nonnegative integer t such that $q^t \equiv -1 \pmod{p}$. Then C is a self-dual code with an automorphism σ of prime order $p \neq char(\mathbb{F}_q)$ if and only if the following two conditions hold.*

(i) $\pi(F_\sigma(C))$ is a self-dual code of length $c+f$ under the inner product (4.8).

(ii) $\varphi(E_\sigma(C)^*)$ is a self-dual code of length c over the field P under the inner product (4.9).

Proof. Assume that C is self-dual. Conditions (i) and (ii) follow from Lemma 4.3.9 and Theorem 4.3.11, respectively. Reciprocally, assume (i) and (ii). In this case, we get that $\dim_{\mathbb{F}_q}(\pi(F_\sigma(C))) = \frac{c+f}{2}$ and $\dim_p(\varphi(E_\sigma(C)^*)) = \frac{c}{2}$. Therefore, $\dim_{\mathbb{F}_q}(E_\sigma(C)) = \dim_{\mathbb{F}_q}(\varphi(E_\sigma(C)^*)) = (p-1)\frac{c}{2}$. Since $C = F_\sigma(C) \oplus E_\sigma(C)$, then $\dim_{\mathbb{F}_q}(C) = \frac{(c+f)}{2} + \frac{c(p-1)}{2} = \frac{(cp+f)}{2} = \frac{n}{2}$. Now let's prove that $C \leqslant C^\perp$. Since $F_\sigma(C) \perp E_\sigma(C)$, it is sufficient to prove that $F_\sigma(C)$ and $E_\sigma(C)$ are self-orthogonal. For $F_\sigma(C)$ the statement is trivial.

Let $a(x) = \alpha_0 + \alpha_1 x + \cdots + \alpha_{p-1}x^{p-1}$, $b(x) = \beta_0 + \beta_1 x + \cdots + \beta_{p-1}x^{p-1} \in \mathcal{P}$. If $a = (\alpha_0, \ldots, \alpha_{p-1})$ and $b = (\beta_0, \ldots, \beta_{p-1})$, then

$$a(x)b(x^{-1}) = (\alpha_0 + \cdots + \alpha_{p-1}x^{p-1})(\beta_0 + \beta_1 x^{p-1} + \cdots + \beta_{p-1}x)$$

$$= a \cdot b + (a \cdot (b\sigma))x + \cdots + (a \cdot (b\sigma^{p-1}))x^{p-1}.$$

For $u = (u_1(x), \ldots, u_c(x))$, $v = (v_1(x), \ldots, v_c(x)) \in \mathcal{P}^c$ we have

$$\sum_{i=1}^{c} u_i(x)v_i(x^{-1}) = \sum_{i=1}^{c} u_i \cdot v_i + (\sum_{i=1}^{c} u_i \cdot (v_i\sigma))x + \cdots + (\sum_{i=1}^{c} u_i \cdot (v_i\sigma^{p-1}))x^{p-1}.$$

Suppose that C_φ is a self-dual code with respect to the Hermitian inner product (4.9). If $u, v \in C_\varphi$ then

$$0 = \langle u, v \rangle = \sum_{i=1}^{c} u_i(x)v_i(x^{-1}) = \sum_{i=1}^{c} u_i \cdot v_i + (\sum_{i=1}^{c} u_i \cdot (v_i\sigma))x + \cdots + (\sum_{i=1}^{c} u_i \cdot (v_i\sigma^{p-1}))x^{p-1}.$$

It turns out that

$$\sum_{i=1}^{c} u_i \cdot v_i = \sum_{i=1}^{c} u_i \cdot (v_i\sigma) = \cdots = \sum_{i=1}^{c} u_i \cdot (v_i\sigma^{p-1}) = 0.$$

If $u_i(x) = u_{i0} + \cdots + u_{i,p-1}x^{p-1}$, $v_i(x) = v_{i0} + \cdots + v_{i,p-1}x^{p-1}$, $i = 1, \ldots, c$, and $u' = (u_{00}, \ldots, u_{c,p-1}) \in \mathbb{F}_q^{pc}$, $v' = (v_{00}, \ldots, v_{c,p-1}) \in \mathbb{F}_q^{pc}$, then $u', v' \in E_\sigma(C)^*$ and

$$u' \cdot v' = \sum_{i=1}^{c} \sum_{j=0}^{p-1} u_{ij} v_{ij} = \sum_{i=1}^{c} u_i \cdot v_i = 0.$$

Hence the codewords of $E_\sigma(C)^*$ are orthogonal to each other and the code is self-orthogonal.

□

The following result is a generalization of (YORGOV, 1987, Theorem 3).

Theorem 4.3.14. *Let C and C' be self-dual codes in $\mathbb{F}_q^n$ and let $\sigma \in PAut(C)$ of prime order $p \neq char(\mathbb{F}_q)$. A sufficient condition for equivalence of C and C' with $\sigma \in PAut(C')$ is that C' can be obtained from C by*

(i) a substitution $x \longmapsto x^t$ in $\varphi(E_\sigma(C)^*)$ where t is an integer with $1 \leqslant t \leqslant p-1$;

(ii) a multiplication of the j-th coordinate of $\varphi(E_\sigma(C)^*)$ by x^{t_j} where t_j is an integer with $0 \leqslant t_j \leqslant p-1$ and $j = 1, \ldots, c$;

(iii) permutation of the first c cycles of C;

(iv) permutation of the last f coordinates of C.

We apply these results to give a classification of all extremal Type I and Type III codes of length 60 with an automorphism of order 29. According to Lemma 4.3.7, we can consider only permutation automorphisms of this order. We focus on permutation σ of type 29-(2,2).

Let C be a binary or ternary self-dual $[60, 30, d > 2]$ code with a permutation automorphism

$$\sigma = (1, 2, \ldots, 29)(30, 31, \ldots, 58). \tag{4.11}$$

By Lemma 4.3.9, $\pi(F_\sigma(C))$ is a self-dual $[4, 2, 2]$ code over $\mathbb{F}_2$ or $\mathbb{F}_3$, respectively, with respect to the inner product (4.8). Thus,

$$\mathrm{gen}(F_\sigma(C)) = \left(\begin{array}{cc|cc} \mathbf{1} & \mathbf{0} & 1 & 0 \\ \mathbf{0} & \mathbf{1} & 0 & 1 \end{array}\right),$$

where $\mathbf{1}$ is the all-ones vector and $\mathbf{0}$ the zero-vector of length 29.

Next we determine $E_\sigma(C)$. Note that $s(q,29) = 28$ and $q^{14} \equiv -1 \pmod{29}$ for $q = 2$ and $q = 3$. Thus, by Theorem 4.3.13, $C_\varphi = \varphi(E_\sigma(C)^*)$ is a self dual $[2,1]$ code over the field $\mathcal{P} = \mathbb{F}_{q^{28}}$ under the Hermitian product

$$\langle u, v\rangle = u_1(x)v_1(x)^{q^{14}} + u_2(x)v_2(x)^{q^{14}}.$$

According to Lemma 4.3.10, the identity element of $\mathcal{P}$ is $e_2(x) = x+x^2+\cdots+x^{28}$ for $q = 2$, and $e_3(x) = 2 + x + x^2 + \cdots + x^{28}$ for $q = 3$. Because of the orthogonality, the weight of all nonzero codewords in C_φ is equal to 2. Hence, $\mathrm{gen}(C_\varphi) = (e(x), a(x))$, where $0 \neq a(x) \in \mathcal{P}$, and $e(x)$ is the identity of $\mathcal{P}$. If α is a primitive element of the field $\mathcal{P}$, we have $a(x) = \alpha(x)^t$ for some t with $0 \leqslant t \leqslant q^{28} - 2$. Due to the orthogonality we get

$$\langle (e(x), a(x)), (e(x), a(x))\rangle = e(x) + a(x)^{q^{14}+1} = e(x) + \alpha(x)^{(q^{14}+1)t} = 0.$$

Then $\alpha(x)^{(q^{14}+1)t} = -e(x)$. Since the order of α is $q^{28} - 1$, we have $t = (2^{14} - 1)k$ in the binary case, $0 \leqslant k \leqslant 2^{14}$, and $t = \dfrac{3^{14} - 1}{2}k$ in the ternary case, $0 \leqslant k \leqslant 2.3^{14} + 1$ with k an odd integer. Let $\delta = \alpha^{2^{14}-1}$ in the binary case, and $\delta = \alpha^{(3^{14}-1)/2}$ in the ternary case, respectively. It follows that $\mathrm{gen}(C_\varphi) = (e(x), \delta^k)$.

Let $c(x) \in \mathcal{P}$, $c(x) = c_0 + c_1 + \cdots + c_{28}x^{28}$. Denote by $[c(x)]$ the 28×29 circulant matrix with first row $(c_0, c_1, \ldots, c_{28})$. From the considered generator matrix of the code C_φ we obtain $\mathrm{gen}(E_\sigma(C)^*) = ([e(x)], [\delta^k])$. So we proved the following lemma.

Lemma 4.3.15. *Let C be a self-dual $[60, 30, d > 2]_q$ code, $q = 2$ or 3, with a permutation automorphism of type 29-$(2,2)$. Let α be a primitive element of the field $\mathcal{P}$, and e be its identity element. Then the code C has a generator matrix in the form*

$$A = \left(\begin{array}{cc|cc} \mathbf{1} & \mathbf{0} & 1 & 0 \\ \mathbf{0} & \mathbf{1} & 0 & 1 \\ \hline [e(x)] & [\delta^k] & 0 & 0 \end{array}\right), \tag{4.12}$$

where $\delta = \alpha^{(q^{14}-1)/(q-1)}$, $0 \leqslant k \leqslant (q-1)q^{14} + q - 2$.

By (MALLOWS; SLOANE, 1973) t is known that the weight enumerator of an extremal $[60, 30, 18]$ Type III code C is given by

$$W_C(y) = \sum_{j=0}^{60} A_j y^j = \sum_{i=0}^{5} a_i f(y)^{15-3i} g(y)^i,$$

where $f(y) = 1 + 8y^3$, $g(y) = y^3(1 - y^3)^3$ and $a_i \in \mathbb{Z}$. A simple calculation shows that $a_0 = 1$, $a_1 = -120$, $a_2 = 4440$, $a_3 = -53320$, $a_4 = 140760$ and $a_5 = -41184$. Therefore, the weight enumerator is uniquely determined and is given by

$$\begin{aligned}
A_{18} &= 3901080 \\
A_{21} &= 241456320 \\
A_{24} &= 8824242960 \\
A_{27} &= 172074038080 \\
A_{30} &= 1850359081824 \\
A_{33} &= 11014750094040 \\
A_{36} &= 36099369380880 \\
A_{39} &= 63958467767040 \\
A_{42} &= 59278900150800 \\
A_{45} &= 27270640178880 \\
A_{48} &= 5739257192760 \\
A_{51} &= 485029078560 \\
A_{54} &= 13144038880 \\
A_{57} &= 71451360 \\
A_{60} &= 41184
\end{aligned}$$

Theorem 4.3.16. *There are exactly three inequivalent extremal* $[60, 30, 18]$ *Type III codes with an automorphism of order* $p = 29$.

Proof. There are two possible types for a permutation automorphism of order 29, either 29-$(1, 31)$ or 29-$(2, 2)$. For the first case, we have that $E_\sigma(C)^*$ is a $[29, 14, d']$ ternary code with $d' \geqslant 18$. However, by (GRASSL, 2007) such a code does not exist and the type of σ is 29-$(2, 2)$. Similarly as in the binary case, we reduce the number of possibilities for the generating matrix (4.12). Now $\mathcal{P}$ is a field with $3^{28} - 1$ elements and $\delta \in \mathcal{P}$ is an element of order $2(3^{14} + 1) = 29 \cdot 329860$. As the binary case, the element $xe(x)$ of order 29 belongs to the cyclic group $\langle\delta\rangle$, and $\gcd(29, 329860) = 1$, so each element of $\langle\delta\rangle$ can be written in the form $x^s\theta^k$, where $\theta \in \langle\delta\rangle$ has order 329860. According to Theorem 4.3.14, we can consider only the elements $a(x) = \theta^k$ for $0 \leqslant k \leqslant 329859$. The transformation $k \longmapsto 3k$ divides the set $\mathbb{Z}_{329860}$ in 11786 orbits $\mathrm{orb}(k)$, of which only 5893 correspond to odd integers k. With MAGMA we have checked that only the values 1031, 2261, 82465, 16493 and 181423

lead to an extremal code. More precisely, we found that the values 181423 and 16493 corresponding to a new code, which we call $\mathcal{V}(29)$. The codes corresponding to 2261 and 1031 are equivalent to $\mathcal{XQR}$ and the code associated to 82465 is $\mathcal{P}(29)$. Using MAGMA one gets the automorphism groups:

- $\mathrm{Aut}(\mathcal{V}(29)) = \mathrm{SL}_2(29)$, $|\mathrm{Aut}(\mathcal{V}(29))| = 24360 = 2^3 \cdot 3 \cdot 5 \cdot 7 \cdot 29$
- $\mathrm{Aut}(\mathrm{XQR}(59)) = \mathrm{SL}_2(59)$, $|\mathrm{Aut}(\mathrm{XQR}(59))| = 205320 = 2^3 \cdot 3 \cdot 5 \cdot 29 \cdot 59$
- $\mathrm{Aut}(\mathcal{P}(29)) = 2.(\mathrm{PSL}_2(59) \times C_4)$, $|\mathrm{Aut}(\mathcal{P}(29))| = 97440 = 2^5 \cdot 3 \cdot 5 \cdot 7 \cdot 29$.

□

Remark 4.3.17. *The primitive element used in Theorem 4.3.16 has the following coefficients*

$$[0, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1].$$

Then we have seen that using the decomposition of self-dual ternary codes is a nice approach that makes the task to classify these codes easier. We were able to obtain a third extremal Type III code of length 60 invariant under $\mathrm{SL}_2(29)$ and a second extremal Type III code of length 52. It is the fact that the automorphism group of $\mathcal{V}(29)$ contains $\mathrm{SL}_2(29)$, that motivated us to try to obtain a family of invariant codes, by means of monomial representations and it is what the final chapter deals with.

4.4 Open question for extremal Type III codes

We have seen how working with the automorphism type of extremal Type III codes, one can classify codes, as was done for the Ternary codes with parameters [60,30,18], [52,26,15] and [36,18,12], all extremal. The classification was done considering an autormorphism of order greater or equal to 5. We would like to investigate if the number of non-equivalent codes known up to this point does not change when considering also automorphisms of order 3 and 2.

5 Generalization of V(29)

5.1 Introduction

In coding theory, it has been always an interesting problem to look for codes with the highest possible minimum distance. Mathematically those objects, if they are also self-dual, then they are of particular interest. This because as shown in (HARADA; KITAZUME; OZEKI, 2002), if these codes are what they call admissible, then there is a construction that leads to an Unimodular lattices. Also, some codes with an automorphism group that contains the special linear group are beautiful examples due to their symmetry. This contents of this section is related to a published work in (NEBE; VILLAR, 2013) and an accepted paper to appear in the Journal Mathematics. Therefore, there are some minor changes in the notation. For instance, instead of $\mathbb{F}_q$ the field is denoted by K; the set of isometries denoted in Chapter 1 by $\mathrm{Isom}(n)$ is denoted here $\mathrm{Mon}_n(K^*)$ and its definition is presented in further detail, since it is in this chapter where this concept is largely used.

One of such a family was discovered in 1969 by Vera Pless (PLESS, 1969), a family of self-dual ternary codes $\mathcal{P}(p)$ of length $2(p+1)$ for primes p with $p \equiv -1 \pmod 6$. Together with the extended quadratic residue codes $\mathrm{XQR}(q)$ of length $q+1$ (q prime, $q \equiv \pm 1 \pmod{12}$) they define a series of self-dual ternary codes of high minimum distance (see (MACWILLIAMS; SLOANE, 1977, Chapter 16, §8)). For $p = 5$, the Pless code $\mathcal{P}(5)$ coincides with the Golay code $\mathfrak{g}_{12}$ which is also the extended quadratic residue code $\mathrm{XQR}(11)$ of length 12.

A. Gleason by means of invariant theory of finite groups, proved that the minimum distance of a self-dual ternary code of length $4n$ cannot exceed $3\lfloor\frac{n}{12}\rfloor + 3$ (GLEASON, 1971). Whenever a self-dual codes reaches the equality, it is called *extremal*. Both constructions, the extended quadratic residue codes and the Pless symmetry codes yield extremal ternary self-dual codes for small values of p.

In this section it is given an interpretation of the Pless codes using monomial representations of the group $\mathrm{SL}_2(p)$. This construction allows to read off a large subgroup

of the automorphism group of the Pless codes (which was already described in (PLESS, 1969)). Also, a related but different series of monomial representations of $\mathrm{SL}_2(p)$ is looked into to construct a new series of self-dual ternary codes $\mathcal{V}(p)$ of length $2(p+1)$, for all primes $p \equiv 5 \pmod 8$, as seen in (NEBE; VILLAR, 2013). The group $\mathrm{SL}_2(p)$ is contained in the automorphism group of $\mathcal{V}(p)$. For $p = 5$ again, as expected due to the classification of self-dual ternary codes given by Mallows, Pless and Sloane in (MALLOWS; PLESS; SLOANE, 1976) we find $\mathcal{V}(5) \cong \mathfrak{g}_{12}$ the Golay code of length 12, but for larger primes these codes are new. In particular, the code $\mathcal{V}(29)$ is an extremal ternary self-dual code, also called extremal *type III code*, of length 60, so we now know three, non-equivalent, extremal ternary codes of length 60: XQR(59), $\mathcal{P}(29)$ and $\mathcal{V}(29)$.

5.2 Codes and monomial groups.

Let K be a field, $n \in \mathbb{N}$. Then the **monomial group** $\mathrm{Mon}_n(K^*) \leqslant \mathrm{GL}_n(K)$ is the group of monomial $n \times n$-matrices over K, where a matrix is called **monomial**, if it contains exactly one non-zero entry in each row and each column. So $\mathrm{Mon}_n(K^*) \cong K^* \wr S_n \cong (K^*)^n : S_n$ is the semidirect product of the subgroup $(K^*)^n$ of diagonal matrices in $\mathrm{GL}_n(K)$ with the group of permutation matrices. For any subgroup $S \leqslant K^*$ we define $\mathrm{Mon}_n(S) := S^n \wr S_n$ to be the subgroup of monomial matrices having all non-zero entries in S. There is a natural epimorphism $\pi : \mathrm{Mon}_n(S) \to S_n$ mapping any monomial matrix to the associated permutation.

By MacWilliam's extension Theorem ((MACWILLIAMS, 1962), see (WARD; WOOD, 1996)) any K-linear weight preserving isomorphism between two subspaces of K^n is the restriction of a monomial transformation in $\mathrm{Mon}_n(K^*)$. This justifies the following commonly used notion of equivalence of codes, which also motivates the investigation of monomial representations of finite groups to find good codes with large automorphism groups.

Definition. 5.2.1. *A K-code C of length n is a subspace of K^n. Given C and C' two codes of length n we say they are* **monomially equivalent**, *if there is some $g \in \mathrm{Mon}_n(K^*)$, such that $Cg = C'$. And we say the* **monomial automorphism group** *of C is the set of such isomorphisms for $C' = C$ denoted by*

$$\mathrm{Aut}(C) := \{g \in \mathrm{Mon}_n(K^*) : Cg = C\}.$$

5.3 Endomorphism rings of monomial representations.

The theory exposed in this section is well known, a nice explicit formulation is contained in (MÜLLER, 2003, Section I (1)).

Let G be some group. A linear K-representation Δ of degree n is a group homomorphism $\Delta : G \to \mathrm{GL}_n(K)$. The representation is called **monomial**, if its image $\Delta(G)$ is conjugate in $\mathrm{GL}_n(K)$ to some subgroup of $\mathrm{Mon}_n(K^*)$.

We call the monomial representation **transitive**, if $\pi(\Delta(G))$ is a transitive subgroup of S_n. In this case the set

$$\{h \in G : 1\pi(\Delta(h)) = 1\} =: H$$

is a subgroup of index n in G and Δ is obtained by inducing up a degree 1 representation of H as follows:

Let H be a subgroup of G of index $n := [G : H]$. Choose $g_1, \ldots, g_m \in G$ such that

$$G = \dot{\cup}_{\ell=1}^{m} Hg_\ell H$$

and put

$$H_\ell := H \cap g_\ell^{-1} H g_\ell.$$

Choose some right transversal $h_{\ell,j}$ of H_ℓ in H, so that $h_{\ell,1} = 1$ and $H = \dot{\cup}_{j=1}^{k_\ell} Hh_{\ell,j}$. Then

$$G = \dot{\cup}_{\ell=1}^{m} \dot{\cup}_{j=1}^{k_\ell} Hg_\ell h_{\ell,j}$$

and the right transversal $\{g_\ell h_{\ell,j} : \ell = 1, \ldots, m, k = 1, \ldots, k_\ell\}$ is a set of cardinality n which we will use as an index set of our $n \times n$-matrices.

For a group homomorphism $\lambda : H \to K^*$ the associated **monomial representation** of G is $\Delta := \lambda_H^G : G \to \mathrm{Mon}_n(\lambda(H))$ defined by

$$(\lambda_H^G(g))_{g_\ell h_{\ell j}, g_{\ell'} h_{\ell', j'}} = \begin{cases} 0 & \text{if } g_\ell h_{\ell j} g (g_{\ell'} h_{\ell', j'})^{-1} \notin H \\ \lambda(g_\ell h_{\ell j} g (g_{\ell'} h_{\ell', j'})^{-1}) & \text{if } g_\ell h_{\ell j} g (g_{\ell'} h_{\ell', j'})^{-1} \in H \end{cases}.$$

The representation λ restricts in two obvious ways to a representation of H_ℓ:

$$\lambda_\ell : H_\ell \to K^*, h \mapsto \lambda(h)$$

and

$$\lambda_\ell^{g_\ell} : H_\ell \to K^*, h \mapsto \lambda(g_\ell h g_\ell^{-1}).$$

Let $\mathcal{I} := \{\ell \in \{1,\dots,m\} : \lambda_\ell = \lambda_\ell^{g_\ell}\}$ and reorder the double coset representatives so that $\mathcal{I} = \{1,\dots,d\}$. Then the **endomorphism ring**

$$\mathrm{End}(\Delta) := \{X \in K^{n\times n} : X\Delta(g) = \Delta(g)X \text{ for all } g \in G\}$$

has dimension d and the **Schur basis** of $\mathrm{End}(\Delta)$ is $(B_1 = I_n, B_2, \dots, B_d)$ where $(B_\ell)_{1,g_\ell} = 1$ and $(B_\ell)_{1,g_k h_{k,i}} \neq 0$ if and only if $\ell = k$. As $\Delta(h_{\ell,k})B_\ell = B_\ell\Delta(h_{\ell,k})$ we conclude

$$\lambda(h_{\ell,k})(B_\ell)_{1,g_\ell h_{\ell,j}} = \Delta(h_{\ell,k})_{g_\ell,g_\ell h_{\ell,j}} = \lambda(h_{\ell,k})\lambda(h_{\ell,j}^{-1}),$$

so $(B_\ell)_{1,g_\ell h_{\ell,j}} = \lambda(h_{\ell,j})^{-1}$ for all j. More generally, we get

Lemma 5.3.1. *$(B_\ell)_{g_k h_{k,i},g_{k'}h_{k',i'}} = 0$ if $g_{k'}h_{k',i'}h_{k,i}^{-1}g_k^{-1} \notin Hg_\ell H$. Otherwise, write for some $h \in H$, $g_{k'}h_{k',i'}h_{k,i}^{-1}g_k^{-1} = hg_\ell h_{\ell,j}$. Then*

$$(B_\ell)_{g_k h_{k,i},g_{k'}h_{k',i'}} = \lambda(h)^{-1}\lambda(h_{\ell,j}^{-1}).$$

Proof. To see this we choose $g = (g_k h_{k,i})^{-1} \in G$. Then $\Delta(g)_{g_k h_{k,i},1} = 1$ and hence

$$(\Delta(g)B_\ell)_{g_k h_{k,i},g_\ell h_{\ell,j}} = \Delta(g)_{g_k h_{k,i},1}(B_\ell)_{1,g_\ell h_{\ell,j}} = \lambda(h_{\ell,j})^{-1}.$$

On the other hand

$$(B_\ell\Delta(g))_{g_k h_{k,i},g_\ell h_{\ell,j}} = (B_\ell)_{g_k h_{k,i},g_{k'}h_{k',i'}}\Delta(g)_{g_{k'}h_{k',i'},g_\ell h_{\ell,j}}$$

for the unique (k',i') such that

$$h := g_{k'}h_{k',i'}(g_k h_{k,i})^{-1}(g_\ell h_{\ell,j})^{-1} \in H$$

and then $\Delta(g)_{g_{k'}h_{k',i'},g_\ell h_{\ell,j}} = \lambda(h)$. As $\Delta(g)B_\ell = B_\ell\Delta(g)$ we compute

$$\lambda(h_{\ell,j})^{-1} = (B_\ell)_{g_k h_{k,i},g_{k'}h_{k',i'}}\lambda(h).$$

□

5.4 Generalized Pless codes.

This section is a reinterpretation of the construction of the famous Pless symmetry codes $\mathcal{P}(p)$ discovered by Vera Pless (PLESS, 1969), (PLESS, 1972b). Explicit

generator matrices for the Pless codes may be obtained from the endomorphism ring of a monomial representation. Let p be an odd prime and

$$\mathrm{SL}_2(p) := \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathbb{F}_p^{2\times 2} : ad - bc = 1 \right\}$$

the group of 2×2-matrices over the finite field $\mathbb{F}_p$ with determinant 1. Let

$$B := \left\{ \begin{pmatrix} a & b \\ 0 & d \end{pmatrix} \in \mathrm{SL}_2(p) \right\} = \left\langle h_1 := \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \zeta := \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^{-1} \end{pmatrix} \right\rangle.$$

Then B is a subgroup of $\mathrm{SL}_2(p)$ or index $p+1$, isomorphic to the semidirect product $C_p : C_{p-1}$, with center

$$Z(B) = Z(\mathrm{SL}_2(p)) = \langle \zeta^{(p-1)/2} \rangle = \{\pm I_2\}.$$

Let

$$\lambda : B \to K^*, \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \mapsto 1, \zeta \mapsto -1$$

Then $\lambda\left(\begin{pmatrix} a & b \\ 0 & d \end{pmatrix}\right) = \left(\frac{a}{p}\right)$ is just the Legendre symbol of the upper left entry. Let

$$\Delta := \lambda_B^{\mathrm{SL}_2(p)} : \mathrm{SL}_2(p) \to \mathrm{Mon}_{p+1}(K^*)$$

be the monomial representation induced by λ. The following facts about this representation are well known, and easily computed from the general description in the previous section.

Remark 5.4.1. *(1) (Gauß-Bruhat decomposition)* $\mathrm{SL}_2(p) = B \dot\cup BwB$, *where*

$$w = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

(2) $B \cap wBw^{-1} = \langle \zeta \rangle$.

(3) A right transversal of B *in* $\mathrm{SL}_2(p)$ *is given by* $[1, wh_x : x \in \mathbb{F}_p]$, *where*

$$h_x := \begin{pmatrix} 1 & 0 \\ x & 1 \end{pmatrix} \in B.$$

(4) The Schur basis of $\mathrm{End}(\Delta)$ *is* (I_{p+1}, P)*, where* $P_{1,1} = 0$*,* $P_{1,wh_x} = 1$ *for all* x*. Then* $P_{wh_x,1} = \left(\frac{-1}{p}\right)$ *and*

$$P_{wh_x,wh_y} = \begin{cases} \left(\frac{x-y}{p}\right) & x \neq y \\ 0 & \boldsymbol{x} = y. \end{cases}$$

(5) $P^2 = \left(\frac{-1}{p}\right) p$ *and* $PP^{tr} = p$*.*

To construct monomial representations of degree $2(p+1)$ the following group is considered

$$\mathcal{G}(p) := \left\langle \begin{pmatrix} \Delta(g) & 0 \\ 0 & \Delta(g) \end{pmatrix}, Z := \begin{pmatrix} 0 & I_{p+1} \\ jI_{p+1} & 0 \end{pmatrix} : g \in \mathrm{SL}_2(p) \right\rangle \leqslant \mathrm{Mon}_{2(p+1)}(K^*),$$

where $j = -\left(\frac{-1}{p}\right) = \begin{cases} 1 & p \equiv 3 \pmod 4 \\ -1 & p \equiv 1 \pmod 4). \end{cases}$

Remark 5.4.2. *Then we can say the following is true.*

(1) $\mathcal{G}(p) \cong \begin{cases} C_4 \times \mathrm{PSL}_2(p) & p \equiv 1 \pmod 4 \\ C_2 \times \mathrm{SL}_2(p) & p \equiv 3 \pmod 4 \end{cases}$

(2) $\mathrm{End}(\mathcal{G}(p)) = \left\{ \begin{pmatrix} A & B \\ jB & A \end{pmatrix} : A, B \in \mathrm{End}(\Delta) \right\}$ *is spanned by*

$$I_{2(p+1)}, X := \begin{pmatrix} P & 0 \\ 0 & P \end{pmatrix}, Y := \begin{pmatrix} 0 & I_{p+1} \\ jI_{p+1} & 0 \end{pmatrix}, XY = \begin{pmatrix} 0 & P \\ jP & 0 \end{pmatrix},$$

such that $X^2 = -jp$*,* $Y^2 = j$*,* $XY = YX$*,* $(XY)^2 = -p$*.*

Definition 5.4.3. *Let* $K = \mathbb{F}_q$ *be the finite field with* q *elements and suppose that there is some* $a \in K^*$ *such that* $a^2 = -p$*. Then put*

$$P_q(p) := aI_{2(p+1)} + XY \in \mathrm{End}(\mathcal{G}(p))$$

and define the **generalized Pless code**

$$\mathcal{P}_q(p) \leqslant K^{2(p+1)}$$

to be the code that is spanned by the rows of $P_q(p)$*.*

Theorem 5.4.4. *Let $a \in \mathbb{F}_q^*$ such that $a^2 = -p$. The code $\mathcal{P}_q(p)$ has generator matrix $(aI_{p+1}|P)$ and is a self-dual code in $\mathbb{F}_q^{2(p+1)}$, with minimum distance $d(\mathcal{P}_q(p)) \leqslant (p+7)/2$ if q is odd and $d(\mathcal{P}_q(p)) \leqslant 4$ if q is even. And its autormorphismgroup $\mathrm{Aut}(\mathcal{P}_q(p))$ contains the subgroup $\mathcal{G}(p)$.*

Proof. The fact that the group $\mathcal{G}(p) \leqslant \mathrm{Aut}(\mathcal{P}_q(p))$ comes out of the construction. As

$$PP^{tr} = pI_{p+1} = -a^2 I_{p+1}$$

the code $\mathcal{P}_q(p)$ is self-dual with respect to the standard inner product.
When adding from the generator matrix $(aI_{p+1}|P)$ the first two rows, the upper bound on the minimum distance is obtained. The sum has weight 4 if q is even. If q is odd then the first row of P is of the form $(0, 1^p)$ and the second row of P is of the form $(-1, 0, v)$, where $v \in \{\pm 1\}^{p-1}$ has exactly $(p-1)/2$ ones and $(p-1)/2$ minus ones. □

Remark 5.4.5. *For $K = \mathbb{F}_3$ and $p \equiv -1 \pmod 3$ we may choose $a = 1$ Then*

$$\mathrm{End}(\mathcal{G}(p)) \cong \begin{cases} \mathbb{F}_9 \oplus \mathbb{F}_9 & p \equiv 1 \pmod 4 \\ \mathbb{F}_3 \oplus \mathbb{F}_3 \oplus \mathbb{F}_3 \oplus \mathbb{F}_3 & p \equiv 3 \pmod 4. \end{cases}$$

The first $p+1$ rows of the rank $p+1$ matrix $I_{2(p+1)} + XY \in \mathrm{End}(\mathcal{G}(p))$ form exactly the generator matrix of $\mathcal{P}_3(p)$, which is the generator matrix for the Pless symmetry code $\mathcal{P}(p)$ as given in (PLESS, 1969).

With the assistance from MAGMA (BOSMA; CANNON; PLAYOUST, 1997) the following invariants of the first few Pless codes are computed:

p	5	11	17	23	29	41	47
$2(p+1)$	12	24	36	48	60	84	96
$d(\mathcal{P}_3(p))$	6	9	12	15	18	21	24
$\mathrm{Aut}(\mathcal{P}_3(p))$	$2.M_{12}$	$\mathcal{G}(11).2$	$\mathcal{G}(17).2$	$\mathcal{G}(23).2$	$\mathcal{G}(29).2$	$\geqslant \mathcal{G}(41)$	$\geqslant \mathcal{G}(47)$

For $q = 5, 7$, and 11 computed $d(\mathcal{P}_q(p))$ with MAGMA in the following scenarios:

(p, q)	$(11, 5)$	$(19, 5)$	$(29, 5)$	$(31, 5)$	$(3, 7)$	$(5, 7)$	$(13, 7)$
$2(p+1)$	12	40	60	64	8	12	28
$d(\mathcal{P}_q(p))$	9	13	18	18	4	6	10

(p,q)	$(17,7)$	$(19,7)$	$(7,11)$	$(13,11)$	$(17,11)$	$(19,11)$
$2(p+1)$	36	40	16	28	36	40
$d(\mathcal{P}_q(p))$	12	13	7	10	12	13

5.5 A new series of self-dual codes invariant under $\mathrm{SL}_2(p)$.

In an analogue manner, applying the same strategy as in Section 5.4, it is now constructed a monomial representation of $\mathrm{SL}_2(p)$ of degree $2(p+1)$, where p is a prime so that $p-1 \equiv 4 \pmod 8$. Here, let's suppose that $\mathrm{char}(K) \neq 2$.

Then the subgroup

$$B^{(2)} := \left\{ \begin{pmatrix} a^2 & 0 \\ b & a^{-2} \end{pmatrix} : a \in \mathbb{F}_p^*, b \in \mathbb{F}_p \right\} \leqslant \mathrm{SL}_2(p)$$

is of index $2(p+1)$ in $\mathrm{SL}_2(p)$ and has a unique linear representation $\gamma : B^{(2)} \to K^*$ with $\gamma(B^{(2)}) = \{\pm 1\}$, then

$$\gamma\left(\begin{pmatrix} a^2 & 0 \\ b & a^{-2} \end{pmatrix}\right) = \left(\frac{a}{p}\right).$$

Thus

$$\Delta' := \gamma_{B^{(2)}}^{\mathrm{SL}_2(p)}$$

is a faithful monomial representation of degree $2(p+1)$.

To obtain explicit matrices let us choose

$$w := \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

as above. By assumption $2 \in \mathbb{F}_p^* \backslash (\mathbb{F}_p^*)^2$ and put

$$\epsilon := \mathrm{diag}(2, 2^{-1}).$$

Then

$$B = B^{(2)} \,\dot\cup\, B^{(2)}\epsilon$$

and

$$\mathrm{SL}_2(p) = B \cup BwB = B^{(2)} \,\dot\cup\, B^{(2)}wB^{(2)} \,\dot\cup\, B^{(2)}\epsilon \,\dot\cup\, B^{(2)}\epsilon wB^{(2)}$$

and a right transversal is given by

$$[1, wh_x, \epsilon, \epsilon wh_x : x \in \mathbb{F}_p],$$

where $h_x := \begin{pmatrix} 1 & 0 \\ x & 1 \end{pmatrix} \in B^{(2)}$.

Lemma 5.5.1. $\mathrm{End}(\Delta')$ *has a Schur basis* $(B_1, B_w, B_\epsilon, B_{\epsilon w} = B_\epsilon B_w)$, *where* B_ϵ, B_w *are given by* $B_\epsilon = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$ *and* $B_w = \begin{pmatrix} X & Y \\ -Y^{tr} & X^{tr} \end{pmatrix}$ *with*

$$X = \begin{pmatrix} 0 & 1 & \dots & 1 \\ -1 & & & \\ \vdots & & R_X & \\ -1 & & & \end{pmatrix}, Y = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & R_Y & \\ 0 & & & \end{pmatrix},$$

in which the rows and columns of R_X *and* R_Y *are indexed by the elements* $\{0, \dots, p-1\}$ *of* $\mathbb{F}_p$ *and*

$$(R_X)_{a,b} = \begin{cases} 0 & b - a \notin (\mathbb{F}_p^*)^2 \\ \left(\frac{c}{p}\right) & b - a = c^2 \in (\mathbb{F}_p^*)^2 \end{cases}$$

$$(R_Y)_{a,b} = \begin{cases} 0 & 2(b - a) \notin (\mathbb{F}_p^*)^2 \\ \left(\frac{c}{p}\right) & 2(b - a) = c^2 \in (\mathbb{F}_p^*)^2 \end{cases}$$

Proof. Explicit computations with the general formulas may be found in Lemma 5.3.1. For instance $(B_w)_{wh_x, wh_y} \neq 0$ if and only if

$$wh_{x-y}w^{-1} = \begin{pmatrix} 1 & y - x \\ 0 & 1 \end{pmatrix} \in B^{(2)} w B^{(2)}.$$

This is equivalent to say that $y - x = a^2$, for some $a \in \mathbb{F}_p$ and then

$$wh_{x-y}w^{-1} = \begin{pmatrix} a^2 & 0 \\ 1 & a^{-2} \end{pmatrix} w \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

and hence $(B_w)_{wh_x, wh_y} = \left(\frac{a}{p}\right)$. □

Remark 5.5.2. *Note that* $(-1) = c^2$ *is a square but not a 4th power, so* $\left(\frac{c}{p}\right) = -1$ *and hence* X *is skew symmetric and*

$$B_w^{tr} = -B_w, B_{\epsilon w}^{tr} = -B_{\epsilon w}.$$

The can be computed that $B_w^2 = B_{\epsilon w}^2 = -p$ *and* $B_\epsilon^2 = -1$ *so*

$$\mathrm{End}(\Delta') \cong \left(\frac{-p,-1}{K}\right)$$

is isomorphic to a quaternion algebra over K*. Also we obtain that*

$$(B_w + B_{\epsilon w})^2 = -2p.$$

Definition. 5.5.1. *Let* p *be a prime* $p \equiv_8 4$*,* $K = \mathbb{F}_q$ *so that there is some* $a \in K^*$ *such that* $a^2 = -tp$ *for* $t = 1$ *or* $t = 2$*. Now let's put*

$$V_t(p) := \begin{cases} aI_{2(p+1)} + B_w, & t = 1 \\ aI_{2(p+1)} + B_w + B_{\epsilon w}, & t = 2 \end{cases}$$

and let $\mathcal{V}_q(p)$ *be the linear code spanned by the rows of* $V_t(p)$*.*

Theorem 5.5.3. $\mathcal{V}_q(p)$ *is a self-dual code in* $\mathbb{F}_q^{2(p+1)}$*. Its monomial automorphism group contains the group* $\mathrm{SL}_2(p)$*.*

Proof. By construction, the code $\mathcal{V}_q(p) \leqslant \mathbb{F}_q^{2(p+1)}$ is invariant under $\mathrm{SL}_2(p) \cong \Delta'(\mathrm{SL}_2(p))$. To see that $\mathcal{V}_q(p)$ is self-orthogonal we check that

$$\begin{aligned} V_1(p)V_1(p)^{tr} &= (a + B_w)(a + B_w^{tr}) \\ &= a^2 + a(B_w + B_w^{tr}) + B_w B_w^{tr} \\ &= a^2 - B_w^2 = 0. \\ V_2(p)V_2(p)^{tr} &= (a + B_w + B_{\epsilon w})(a + B_w^{tr} + B_{\epsilon w}^{tr}) \\ &= a^2 - (B_w + B_{\epsilon w})^2 = 0. \end{aligned}$$

To obtain the rank of the matrix $V_t(p)$ note that

$$\mathrm{End}(\Delta') \cong \left(\frac{-p,-1}{\mathbb{F}_q}\right) \cong \mathbb{F}_q^{2\times 2}.$$

So the representation Δ' is the sum of two equivalent representations over $\mathbb{F}_q$. Which have the same degree, $p+1$, half of the degree of Δ' and therefore $p+1$ divides the rank of any matrix in $\mathrm{End}(\Delta')$. □

Remark 5.5.4. *The matrices of rank* $p+1$ *in* $\mathrm{End}(\Delta')$ *yield* $q+1$ *different self-dual codes invariant under* $\Delta'(\mathrm{SL}_2(p))$*. In general, these fall into different equivalence classes. For instance for* $q = 7$*, where* 2 *is a square mod* 7*, the codes spanned by the rows of* $V_1(p)$ *and* $V_2(p)$ *are nonequivalent. This is also true for* $p = 5$ *and* $p = 13$*, although they have the same minimum distance. For* $q = 3, p = 29$ *all* 4 *codes are equivalent and are just represented as the code* $\mathcal{V}_3(29)$*.*

The first few codes $\mathcal{V}_3(p)$ have the following parameters (computed with MAGMA (BOSMA; CANNON; PLAYOUST, 1997)):

p	5	13	29	37	53
$2(p+1)$	12	28	60	76	108
$d(\mathcal{V}_3(p))$	6	9	18	18	24
$\mathrm{Aut}(\mathcal{V}_3(p))$	$2.M_{12}$	$\mathrm{SL}_2(13)$	$\mathrm{SL}_2(29)$	$\geqslant \mathrm{SL}_2(37)$	$\geqslant \mathrm{SL}_2(53)$

For $q = 5, 7$, and 11 and small lengths we computed $d(\mathcal{V}_q(p))$ with MAGMA:

(p, q)	$(13, 5)$	$(29, 5)$	$(5, 7)$	$(13, 7)$	$(5, 11)$	$(13, 11)$
$2(p+1)$	28	60	12	28	12	28
$d(\mathcal{V}_q(p))$	10	16	6	9	7	11

Here we notice that even though the family yields extremal self-dual codes for $q = 3$ and smaller values of primes p, such that $p \equiv 5 \pmod 8$, the minimum distance does not grow always with p, and for $q > 3$ the minimum distance is also not bigger either. This is even more noticeable for Generalized the Pless Codes presented in Section 5.4, which could be a reason why, if explored by Dr. Pless, she did not present the construction for other values of q other than 3.

On a final note we can clearly determine that the new $\mathcal{V}_3(29)$ is not equivalent to the previously known XQR(59), $\mathcal{P}(29)$ codes, since their automorphism groups are different, which by the way can also be directly verified using MAGMA. Unfortunately when checking the conditions given in (HARADA; KITAZUME; OZEKI, 2002) this new code of length 60 is not admissible and, thus, does not lead to an unknown unimodular lattice either.

All the files corresponding to the algorithms used to classify codes using spectrum, to check the types and look for all the codes with specific parameters, as well as, the output files are contained in GitHub accessible via Darwin Villar's GitHub repository.

5.6 Open question on the generalization of $\mathcal{V}_3(29)$

We found that the new extremal Type III code of length 60 had and automorphism group containing $\mathrm{SL}_2(29)$ and by means of monomial representations of this group we were able to obtain a family for which this code belonged. We would like to consider

the construction taking into account an hermitian product and not only the euclidean one. Then we would like to analyse if this would produce a new family of hermitian codes.

Bibliography

ASSMUS, E. J.; MATTSON, H. J.; TURYN, R. *Research to Develop the Algebraic Theory of Codes. Applied Research Laboratory, Sylvania Electronic Systems, June 1967. No.* [S.l.], 1967. Cited on page 66.

AZAR, G.; ALAJAJI, F. When are maximum likelihood and minimum distance decoding equivalent for binary contagion channels? In: IEEE. *2013 13th Canadian Workshop on Information Theory.* [S.l.], 2013. p. 12–16. Cited on page 21.

BONFERRONI, C. Teoria statistica delle classi e calcolo delle probabilit á. *Pubblicazioni del R. Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, v. 8, p. 3–62, 1936. Cited on page 35.

BOSMA, W.; CANNON, J.; PLAYOUST, C. The magma algebra system i: The user language. *Journal of Symbolic Computation*, Elsevier, v. 24, n. 3-4, p. 235–265, 1997. Cited 2 times on page 96 and 100.

BOUYUKLIEVA, S.; CRUZ, J. de la; VILLAR, D. Extremal binary and ternary codes of length 60 with an automorphism of order 29 and a generalization. *Mathematics*, MDPI, v. 10, n. 5, p. 748, 2022. Cited 3 times on page 14, 65, and 81.

CHEON, E. Equivalence of linear codes with the same weight enumerator. *Scientiae Mathematicae japonicae*, Japanese Association of Mathematical Sciences, v. 64, n. 1, p. 163, 2006. Cited on page 48.

DIDIER, F. A new upper bound on the block error probability after decoding over the erasure channel. *IEEE Transactions on Information Theory*, IEEE, v. 52, n. 10, p. 4496–4503, 2006. Cited 2 times on page 13 and 22.

FASHANDI, S.; GHARAN, S. O.; KHANDANI, A. K. Coding over an erasure channel with a large alphabet size. In: IEEE. *2008 IEEE International Symposium on Information Theory.* [S.l.], 2008. p. 1053–1057. Cited 2 times on page 13 and 22.

FIRER, M. Alternative metrics. In: HUFFMAN, W. C.; KIM, J.-L.; SOLÉ, P. (Ed.). *Concise Encyclopedia of Coding Theory.* [S.l.]: CRC Press, 2021. cap. 22. Cited on page 21.

GABORIT, P.; OTMANI, A. Experimental constructions of self-dual codes. *Finite Fields and Their Applications*, Elsevier, v. 9, n. 3, p. 372–394, 2003. Cited 2 times on page 69 and 72.

GLEASON, A. M. Weight polynomials of self-dual codes and the macwilliams identities. In: GAUTHIER-VILLARS. *Actes Congres Int. de Mathematique, 1970.* [S.l.], 1971. Cited 2 times on page 66 and 90.

GOLAY, M. J. Notes on digital coding. *Proc. IEEE*, v. 37, p. 657, 1949. Cited on page 66.

GRASSL, M. Code tables: Bounds on the parameters of various types of codes. *http://www. codetables. de/*, 2007. Cited 6 times on page 65, 70, 71, 72, 77, and 88.

HAMMING, R. W. Error detecting and error correcting codes. *The Bell system technical journal*, Nokia Bell Labs, v. 29, n. 2, p. 147–160, 1950. Cited on page 66.

HARADA, M.; KITAZUME, M.; OZEKI, M. Ternary code construction of unimodular lattices and self-dual codes over $\mathbb{Z}_6$. *Journal of Algebraic Combinatorics*, Springer, v. 16, n. 2, p. 209–223, 2002. Cited 2 times on page 90 and 100.

HELLESETH, T.; KLØVE, T.; YTREHUS, Ø. Generalized hamming weights of linear codes. *IEEE transactions on information theory*, IEEE, v. 38, n. 3, p. 1133–1140, 1992. Cited on page 46.

HUFFMAN, W. Automorphisms of codes with applications to extremal doubly even codes of length 48. *IEEE Transactions on Information Theory*, IEEE, v. 28, n. 3, p. 511–521, 1982. Cited 2 times on page 83 and 84.

_____. Decomposing and shortening codes using automorphisms (corresp.). *IEEE transactions on information theory*, IEEE, v. 32, n. 6, p. 833–836, 1986. Cited 2 times on page 83 and 84.

HUFFMAN, W. C. On extremal self-dual ternary codes of lengths 28 to 40. *IEEE transactions on information theory*, IEEE, v. 38, n. 4, p. 1395–1400, 1992. Cited 2 times on page 71 and 82.

HUFFMAN, W. C.; PLESS, V. *Fundamentals of error-correcting codes*. [S.l.]: Cambridge university press, 2010. Cited on page 15.

JOSHI, K. D. *Foundations of discrete mathematics*. [S.l.]: New Age International, 1989. 141–142 p. Cited on page 15.

LEMES, L. C.; FIRER, M. Generalized weights and bounds for error probability over erasure channels. In: IEEE. *2014 Information Theory and Applications Workshop (ITA)*. [S.l.], 2014. p. 1–8. Cited on page 23.

LING, S.; SOLÉ, P. On the algebraic structure of quasi-cyclic codes. i. finite fields. *IEEE Transactions on Information Theory*, IEEE, v. 47, n. 7, p. 2751–2760, 2001. Cited on page 84.

MACWILLIAMS, F. *Combinatorial properties of elementary Abelian groups Ph. D.* Tese (Doutorado) — thesis, Radcliffe College, 1962. Cited on page 91.

MACWILLIAMS, F. J.; ODLYZKO, A. M.; SLOANE, N. J.; WARD, H. N. Self-dual codes over gf (4). *Journal of Combinatorial Theory, Series A*, Elsevier, v. 25, n. 3, p. 288–318, 1978. Cited on page 66.

MACWILLIAMS, F. J.; SLOANE, N. J. A. *The theory of error correcting codes*. [S.l.]: Elsevier, 1977. v. 16. Cited on page 90.

MALLOWS, C.; PLESS, V.; SLOANE, N. Self-dual codes over gf(3). *SIAM Journal on Applied Mathematics*, SIAM, v. 31, n. 4, p. 649–666, 1976. Cited on page 91.

MALLOWS, C. L.; SLOANE, N. J. An upper bound for self-dual codes. *Information and Control*, Elsevier, v. 22, n. 2, p. 188–200, 1973. Cited 2 times on page 66 and 87.

MÜLLER, J. On endomorphism rings and character tables. Citeseer, 2003. Cited on page 92.

NEBE, G. On extremal self-dual ternary codes of length 48. *International Journal of Combinatorics*, Hindawi, v. 2012, 2012. Cited on page 83.

NEBE, G.; VILLAR, D. An analogue of the Pless symmetry codes. In: BULGARIAN ACADEMY OF SCIENCES. *Seventh International Workshop on Optimal Codes and Related Topics*. [S.l.], 2013. p. 158–163. Cited 4 times on page 14, 65, 90, and 91.

PLESS, V. On a new family of symmetry codes and related new five-designs. *Bulletin of the American Mathematical Society*, American Mathematical Society, v. 75, n. 6, p. 1339–1342, 1969. Cited 5 times on page 69, 90, 91, 93, and 96.

_____. A classification of self-orthogonal codes over gf (2). *Discrete Mathematics*, Elsevier, v. 3, n. 1-3, p. 209–246, 1972. Cited on page 27.

_____. Symmetry codes over gf (3) and new five-designs. *Journal of Combinatorial Theory, Series A*, Elsevier, v. 12, n. 1, p. 119–142, 1972. Cited on page 93.

RAVAGNANI, A. Generalized weights: An anticode approach. *Journal of Pure and Applied Algebra*, Elsevier, v. 220, n. 5, p. 1946–1962, 2016. Cited on page 48.

ROMAN, S. *Coding and information theory*. [S.l.]: Springer Science & Business Media, 1992. v. 134. 69-115 p. Cited on page 21.

ROSENTHAL, J.; YORK, E. V. On the generalized hamming weights of convolutional codes. *IEEE Transactions on Information Theory*, IEEE, v. 43, n. 1, p. 330–335, 1997. Cited 2 times on page 13 and 48.

SHANNON, C. E. Communication in the presence of noise. *Proceedings of the IEEE*, v. 72, n. 9, p. 1192–1201, 1984. Cited on page 21.

SHEN, L.-Z.; FU, F.-W. The decoding error probability of linear codes over the erasure channel. *IEEE Transactions on Information Theory*, IEEE, v. 65, n. 10, p. 6194–6203, 2019. Cited 2 times on page 13 and 46.

SHENGYUAN, Z. On the nonexistence of extremal self-dual codes. *Discrete applied mathematics*, Elsevier, v. 91, n. 1-3, p. 277–286, 1999. Cited on page 65.

TAYLOR, D. E. *The geometry of the classical groups*. [S.l.]: Heldermann Verlag, 1992. v. 9. Cited on page 78.

VILLAR, D. On extremal Type III codes. *http://alcoma15.uni-bayreuth.de/files/slides/contributed/villar.pdf*, 2015. Cited on page 65.

_____. On fourth-power residue self-dual codes. *https://bit.ly/3rxk5kT*, 2015. Cited on page 65.

WARD, H. N.; WOOD, J. A. Characters and the equivalence of codes. *Journal of Combinatorial Theory, Series A*, Academic Press, v. 73, n. 2, p. 348–352, 1996. Cited on page 91.

WEI, V. K. Generalized Hamming weights for linear codes. *IEEE Transactions on Information Theory*, IEEE, v. 37, n. 5, p. 1412–1418, 1991. Cited 6 times on page 13, 15, 18, 19, 20, and 48.

YORGOV, V. Binary self-dual codes with automorphisms of odd order. *Problemy Peredachi Informatsii*, Russian Academy of Sciences, Branch of Informatics, Computer Equipment and . . . , v. 19, n. 4, p. 11–24, 1983. Cited on page 84.

_____. A method for constructing inequivalent self-dual codes with applications to length 56. *IEEE transactions on information theory*, IEEE, v. 33, n. 1, p. 77–82, 1987. Cited on page 86.